Institut Puig Castellar
Santa Coloma de Gramenet

# PROYECTO FINAL DE CURSO

**(Proyecto de investigación)**
2SMXA

**CFGM - Sistemes Microinformàtics i Xarxes**
**Curs: 2024 - 2025**
**MP12 - Crèdit de síntesi (Proyecto final)**
**Rafael Campos Vallone**
**Ivan Padilla Reina**
**Luis Antonio Quiroga Flores**

# Memoria Técnica

**Ciclo Formativo de Grado Medio – Sistemas Microinformáticos y Redes**

Curso 2024-2025

Autores:

- Rafael Campos Vallone
- Iván Padilla Reina
- Luis Antonio Quiroga Flores

## Licencias:



# Resumen del proyecto:

## Temática del proyecto.

Nuestra temática se basará en la ciberseguridad ya que haremos un honeypot, y daremos varios servicios de red.

## Objetivo del proyecto:

Obtener experiencia en búsqueda de información y resolución de problemas a la hora de hacer un proyecto a partir de una idea , es decir, que seamos capaces de conseguir realizar una tarea mediante nosotros mismos para demostrar nuestras capacidades, en este caso demostraremos esto mediante el honeypot.

## *Metodología seguida para conseguir el objetivo.*

Hemos elegido la metodología LEAN porque nos ayuda a trabajar de forma eficiente, optimizando recursos y centrando nuestros esfuerzos en lo importante.

1. **Identificar valor**: Definiremos qué es lo esencial en el proyecto.
2. **Eliminar desperdicios**: Nos enfocaremos solo en tareas necesarias.
3. **Flujo continuo**: Organizaremos el trabajo por fases claras.
4. **Mejora continua**: Revisaremos y mejoraremos constantemente nuestro avance.

Con LEAN, seremos más organizados, aprenderemos en el proceso y aseguraremos un buen resultado en nuestro proyecto.

## Resumen de las conclusiones.

Al elegir una idea que no sabemos si funcionará, como el honeypot, hemos aprendido a trabajar con incertidumbre. Sabemos que esto es parte del proceso y que nos ayudará a mejorar nuestras habilidades para buscar soluciones y adaptarnos. Si vemos que el honeypot no funciona como esperábamos, estamos listos para cambiar de idea o ajustar el proyecto, siempre buscando algo que nos permita avanzar y cumplir nuestros objetivos. Lo importante es aprender, probar cosas nuevas y no tener miedo de cambiar de plan si es necesario.

Palabras clave

- Prototipo
- Seguridad
- Investigación
- Servicios

# Abstract:

## *Project theme.*

Our theme will be based on cybersecurity since we will make a honeypot, and we will provide several network services.

## *Project objective:*

Obtain experience in information search and problem solving when making a project from an idea, that is to say, that we are able to perform a task by ourselves to demonstrate our capabilities, in this case we will demonstrate this through the honeypot.

## *Methodology followed to achieve the objective.*

We have chosen the LEAN methodology because it helps us to work efficiently, optimizing resources and focusing our efforts on what is important.

1. **Identify value:** We will define what is essential in the project.
2. **Eliminate waste:** We will focus only on necessary tasks.
3. **Continuous flow:** We will organize the work by clear phases.
4. **Continuous improvement:** We will constantly review and improve our progress.

With LEAN, we will be more organized, we will learn in the process and we will ensure a good result in our project.

## Summary of conclusions.

By choosing an idea that we don't know if it will work, such as the honeypot, we have learned to work with uncertainty. We know that this is part of the process and that it will help us improve our skills in finding solutions and adapting.

If we see that the honeypot does not work as we expected, we are ready to change the idea or adjust the project, always looking for something that will allow us to move forward and meet our goals. The important thing is to learn, try new things and not be afraid to change the plan if necessary.

## Keywords:

- Prototype
- Security
- Research
- Services

# ÍNDICE

# 1 Introducción

Este documento constituye la memoria técnica del proyecto final del CFGM en Sistemas Microinformáticos y Redes. El trabajo ha sido desarrollado por tres estudiantes durante el curso 2024-2025 y se titula **M.A.S.T.**

El proyecto consiste en el diseño e implementación de un entorno simulado de ciberseguridad basado en **honeypots de alta interacción**. Además, se complementa con la creación de una página web corporativa que ofrece servicios informáticos. De esta forma, se abordan áreas técnicas clave del ciclo: redes, servicios, automatización, diseño web y seguridad.

## Descripción de los integrantes:

**Ivan** →Mi nombre es Iván Padilla Reina, tengo 17 años y actualmente estoy cursando el segundo año de un ciclo formativo de grado medio de Sistemas Microinformáticos y Redes. Ahora voy a explicar el motivo por el cual decidí seguir esta formación. Cuando estaba en 4.º de la ESO, inicialmente tenía pensado realizar el bachillerato. Sin embargo, tras asistir a unas charlas informativas en las que se recomendaban los ciclos formativos como una alternativa práctica y enfocada al mundo laboral, decidí replantearme mi camino académico. Reflexioné sobre mis intereses y decidí que sería buena idea comenzar directamente en el ámbito que realmente me apasiona → la informática. Desde pequeño he sentido una gran curiosidad por la tecnología, una afición que nació gracias a la influencia de mi padre. Con él aprendí los conceptos básicos de informática; juntos manipulábamos consolas, desmontabamos pc'sy realizábamos tareas similares que despertaron en mí el deseo de comprender cómo funcionan las cosas a nivel técnico. Esa motivación me llevó a elegir este ciclo, en el que actualmente me siento muy cómodo, aprendiendo y disfrutando del proceso formativo. En el futuro, mi intención es continuar con un ciclo formativo de grado superior y, posteriormente, acceder a una carrera universitaria en Ingeniería Informática.

**Luis** → Hola, me llamo Luis Antonio Quiroga Flores, tengo 17 años y actualmente estoy cursando el segundo año del grado medio de "Sistemas Microinformatics en Xarxa". Elegí este ciclo porque me interesa la parte de la tecnología/Informática y cómo funciona todo detrás de las pantallas que usamos a diario. Además, me motiva aprender a resolver problemas y encontrar soluciones prácticas, y yo creo

que este camino me ayudará a alcanzar mis metas profesionales en el futuro. Y mi objetivo es adquirir una base sólida que me permita en el futuro trabajar para proyectos relacionados con la tecnología/informática.

**Rafa** → Me llamo Rafael Campos Vallone, tengo 18 años y ahora mismo estoy cursando el segundo año de ciclo de grado medio de “Sistemes Microinformatics en Xarxa”. Escogí seguir estudiando en este grado medio debido a que de pequeño me apasionaban los ordenadores, me encantaba trastear con ellos, desmontarlos, volverlos a montar, cambiarle piezas, mirar el funcionamiento, el hecho de poder jugar a videojuegos y el cómo era posible que desde algo que aparentemente no tiene nada, aparecer personajes, habilidades, paisajes, letras entre muchas otras cosas, todo eso para mi era algo que debía tener alguna explicación, quiero decir después de todo no es magia aunque por aquel entonces si pensaba que todo funcionaba por arte de magia, aunque ahora gracias a este curso comienzo a comprender la complejidad que hay detrás de solo una pantalla.

## ¿Qué pretendemos hacer como proyecto de final de curso?

Bueno, nos gusta el ámbito de la ciberseguridad y el de las redes por lo tanto hemos decidido enfocarnos en esos dos principalmente, pese a que debido a requisitos presentados por nuestros tutores debemos hacer/tocar un poco como mínimo de todos los ámbitos que hemos ido haciendo a lo largo de este ciclo formativo.

En base a eso con unos puntos principales empezamos a plantearnos ideas que creemos que podríamos hacer, entre ellas pensamos hacer un “firewall”, ya que demostraría nuestro conocimiento en ciberseguridad y en redes ya que lo implementaremos a un router virtual para posteriormente ponerlo en práctica, sin embargo cuando planteamos nuestra idea al tutor de ciberseguridad (en este caso Jordi Farrero) nos dijo que crear un firewall y implementarlo era bastante complejo y que se necesita bastante conocimiento y tiempo para poder llevarlo a cabo. Por otro lado nuestro tutor de proyecto (en este caso Fede) nos dijo que crear un firewall sería un trabajo relativamente sencillo y que necesitariamos hacer algo más complejo y o extenso además de crear algo de todas las materias del ciclo.

Una vez habiendo hablado con los profes sobre nuestras ideas para el proyecto, viendo que sus perspectivas eran opuestas decidimos optar por descartar la idea, después de eso pensamos en hacer un pen autoinyectable con un script en bash que al conectarlo al puerto de un dispositivo, recabar la información del dispositivo y lo guardará en el mismo usb sin que el dueño del dispositivo se diera cuenta, para esto planteamos un escenario en el que se podría utilizar esta herramienta (todo esto es solo con fines educativos y no para usarlos de forma maliciosa).

Como no nos acabó de convencer la idea del pen, ya que no sabemos mucho del tema, decidimos tomarnoslo con más calma e ir paso a paso, entonces lo que hemos hecho ha sido una lluvia de ideas, en el cual hemos sacado varias ideas que nos llaman la atención:

## Lluvia de ideas:

- Montar un Honeypot básico.
- Análisis de malware básico en un entorno controlado.
- Escaneo de vulnerabilidades usando herramientas de código abierto.
- Autenticación multifactor, MFA (Multi Factor Authentication) con autenticación basada en TOTP.
- Desarrollo de un sistema de copias de seguridad automatizado y seguro
- Implementar un servidor de VPN (Virtual Private Network).
- Sistema de detección de intrusos básico, IDS (Intrusion Detection System).
- Simulación de ataques.
- Cifrado de archivos y comunicaciones.

Entonces por lo que nos hemos decantado viendo los consejos de los profes y viendo los requisitos del proyecto además de la lluvia de ideas, hemos decidido crear una empresa para cubrir las asignaturas de (EIE y FOL) donde nosotros podamos ofrecer nuestros servicios de seguridad informática (Ciberseguridad), en una página web (Aplicaciones Web y Ofimática) entre otras cosas aunque aún estamos debatiendo ideas.

## 1.1 Contexto

Hasta ahora hemos hecho parte del honeypot, también estamos pensando cómo será nuestra página web para ofrecer nuestros servicios y hemos acabado la tabla de riesgos.

## 1.2 Justificación

Elegimos esta temática porque la ciberseguridad representa un área crítica y en constante evolución dentro del mundo informático. Nos permite adquirir competencias que no se desarrollan en profundidad en el aula, enfrentándonos a situaciones reales como ataques automatizados, gestión de logs y análisis de vulnerabilidades.

Además, al presentar el proyecto como si fuéramos una empresa de servicios TIC, podemos trabajar también la parte emprendedora y de comunicación técnica, cumpliendo así con los objetivos de asignaturas como FOL y EIE.

## 1.3 Objetivos

El objetivo principal por el momento es hacer un honeypot funcional, una página web en la que mostraremos nuestros servicios y demostrar nuestra investigación con hechos y no solo con teoría.

### 1.3.1 Objetivo general

Desarrollar un proyecto técnico funcional que permita aplicar e integrar los conocimientos adquiridos durante el ciclo formativo, aportando una solución práctica centrada en la ciberseguridad y los servicios de red.

### 1.3.2 Objetivos específicos

- Implementar un honeypot completo mediante la solución T-Pot.
- Crear una web profesional de presentación y captación de clientes.
- Configurar un entorno de red con servidores DNS y DHCP.
- Automatizar procesos mediante la herramienta Ansible.
- Monitorizar la red y los servicios con Grafana, Prometheus y Loki.
- Evaluar y documentar el proceso técnico completo.

## 1.4 Estratègia y planificación del proyecto

La estrategia es dividirnos las tareas de forma equitativa para buscar información y luego poder llevar a cabo nuestros objetivos de forma práctica.

| Fase | Descripción | Herramientas/Temas |
|---|---|---|
| 1 | Investigación previa | T-Pot, honeypots. |
| 2 | Instalación de servicios base | Ubuntu Server, DNS, DHCP |
| 3 | Desarrollo web | WordPress, Elementor, Kubio |
| 4 | Honeypot T-Pot | Docker, Kibana, ELK Stack |
| 5 | Monitorización y métricas | Grafana, Prometheus, Loki |
| 6 | Automatización | Ansible, YAML |
| 7 | Documentación y entrega | Word, PDF, capturas de pantalla |

## 1.5 Metodología de trabajo

Hemos decidido que vamos a trabajar con la metodología LEAN, esta tiene como objetivo la optimización de los procesos para alcanzar las metas de una forma más eficiente. Es decir, aprovechando de mejor manera los recursos disponibles, ayudando a identificar y eliminar los retrasos y otras ineficiencias para ofrecer valor más rápido.

Hemos elegido esta metodología ya que nos gustó cómo se organizaba y cómo funcionaba, por eso decidimos seguir esta metodología.

## 1.6 Estudio econòmico y presupuestario

Para realizar ese inventario de las tareas que tenemos que hacer deberemos preguntaros varias cosas, que tan grande va a ser, ya que contra mas grande, mas ordenadores tendremos que poner, o al menos máquinas virtuales.

# Pàgina Web

Nuestra idea principal era utilizar WordPress junto con el plugin Elementor, ya que durante las prácticas de la asignatura de Aplicaciones Web nos pareció una herramienta fácil de usar, eficiente y visualmente atractiva. Sin embargo, al avanzar en el desarrollo, nos dimos cuenta de que no era suficiente para lograr el nivel de personalización que queríamos en nuestra página.

Investigamos distintas opciones para ampliar las funcionalidades del sitio web y descubrimos una gran variedad de plugins disponibles. Tras analizar varias alternativas, decidimos implementar algunos adicionales que ofrecieran más opciones de diseño. Finalmente, incorporamos los plugins **Kubio** y **Spexo**, ambos con versiones de pago similares a Elementor. No obstante, optamos por utilizar únicamente las versiones gratuitas, ya que consideramos que era posible lograr un buen resultado sin necesidad de invertir dinero.

Gracias a estos plugins y complementos, pudimos continuar el desarrollo de nuestro sitio web con un mayor grado de personalización. Sin embargo, también enfrentamos algunas limitaciones técnicas. Uno de los principales inconvenientes fue que no es posible combinar funcionalidades de los distintos editores; por ejemplo, si te gusta cómo Elementor permite insertar imágenes, esa opción no está disponible al trabajar con Kubio.

Además, al alternar entre plugins para realizar distintas tareas, a veces los cambios no se guardan correctamente debido a problemas de compatibilidad. Por esta razón, aprendimos que es fundamental establecer un orden de trabajo para evitar conflictos y pérdida de elementos en la edición.

En este proyecto hemos aplicado todo lo aprendido durante la asignatura, integrando conocimientos técnicos con herramientas prácticas para construir una página web funcional y atractiva.

## **Mysql**:

CREATE DATABASE mast;
CREATE USER 'admin'@'localhost' IDENTIFIED BY 'admin';
GRANT ALL ON mast.* to 'admin'@'localhost';
exit

```
# mast.local.conf
<VirtualHost *:80>
    ServerAdmin admin@mast.local
    ServerName www.mast.local
    ServerAlias mast.local
    DocumentRoot /var/www/mast.local
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

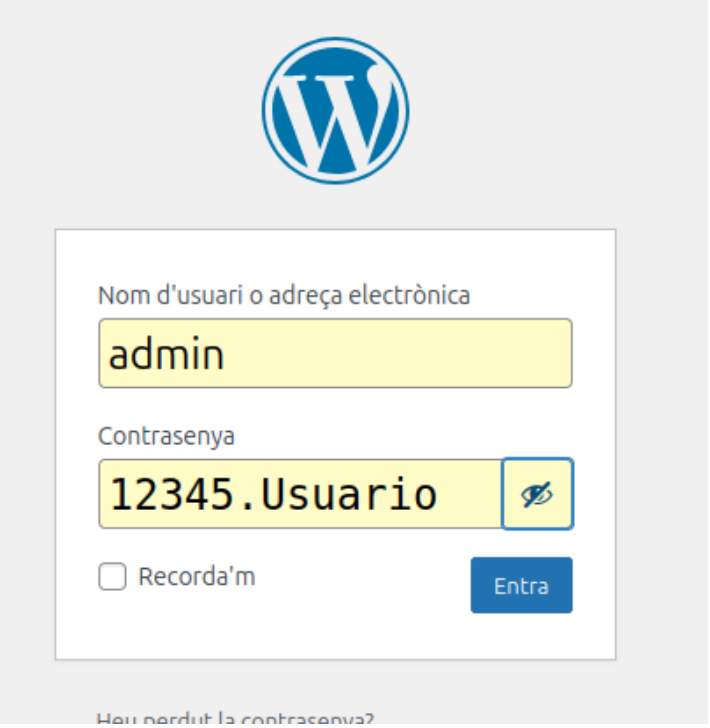
Fichero virtual host

Figura 5: Archivo de configuración del VirtualHost en Apache para el dominio mast.local.

Cuenta gmail:

Correo electrónico:
mast.company.project@gmail.com
Contraseña:
12345.Usuario

Cuenta de Wordpress como administrador para configurar el wordpress:




inicio de sesión wordpress

**Ip de la pagina: 192.168.12.250**

url:

http://192.168.12.250/wp-login.php?redirect_to=http%3A%2F%2F192.168.12.250%2Fwp-admin%2F&reauth=1

## Diseño de la página web:

- **Color**: El colo que hemos elegido para la pagina web es una mezcla de colores entre un verde metalico con azul marino, admas de que tiene toques blancos y negros para poder resaltar algunas partes.

- **Contenido**: Nuestro contenido se basa básicamente en nuestros servicios, es decir, todo lo que ofrecemos en nuestra empresa, destacamos nuestro producto estrella, y brindamos una tabla de precios aproximada donde se puede ver cuánto puede valer un proyecto, además de que incorporamos un

blog donde vamos subiendo información o curiosidad del mundo de la informática y de la ciberseguridad.

- **Apartados**: Nuestra página web está dividida en diferentes apartados:

    - Un apartado donde damos una breve información de todos los servicio que tenemos actualmente en la empresa
    - Un apartado en específico de cada servicio, en estos apartados es exclusivamente de ese servicio y damos una información específica de dicho servicio
    - Un apartado donde nos presentamos y hablamos un poco sobre nosotros, además de que explicamos un poco nuestro origen y desarrollo.
    - Además tenemos un apartado para que los clientes puedan contactar con nosotros por si quieren contratar algún servicio o por si tienen alguna duda en especial.
    - Finalmente tenemos un apartado de blog donde subimos pro tips sobres la ciberseguridad y sobretodo recursos para tu dia a dia

# Servidor DHCP y DNS

**Kea** es un **servidor DHCP (Dynamic Host Configuration Protocol)** de código abierto, desarrollado y mantenido por el **Internet Systems Consortium (ISC)**. Kea está diseñado para reemplazar al clásico **ISC DHCP Server**, ofreciendo una arquitectura modular, alto rendimiento y capacidad de configuración dinámica mediante API RESTful.


---

## ¿Para qué se utiliza Kea DHCP?

Kea DHCP se utiliza para asignar automáticamente direcciones IP y otra información de configuración de red a dispositivos clientes (hosts) en redes IPv4 e IPv6. Tiene tres componentes principales:

- **kea-dhcp4**: Servidor DHCP para redes IPv4.
- **kea-dhcp6**: Servidor DHCP para redes IPv6.
- **kea-ctrl-agent**: Agente de control REST API para gestionar Kea dinámicamente.

También incluye módulos opcionales como:

- **Kea DHCPSA**: Asignación de direcciones basadas en base de datos (MySQL, PostgreSQL).
- **Hooks**: Extensiones para personalizar el comportamiento del servidor.
- **High Availability**: Configuración en clúster activo/pasivo o activo/activo.

---

## ¿Por qué hemos escogido Kea en lugar de otros servidores DHCP?

Hemos optado por **Kea DHCP** en lugar de otros servidores (como ISC DHCP clásico, dnsmasq o Windows DHCP Server) por varias razones técnicas y operativas, además de que es el servicio que hemos dado durante la unidad formativa con Victor Carceler:

### Ventajas de Kea sobre otros servidores DHCP:

| Ventaja | Descripción |
|---|---|
| API RESTful | Permite gestionar configuraciones sin reiniciar el servicio. |
| Modularidad | Componentes desacoplados y extensibles según necesidades. |
| Alto rendimiento | Mejor eficiencia que el servidor ISC DHCP clásico. |
| Backend en base de datos | Soporte nativo para MySQL, PostgreSQL o Cassandra. |
| Seguridad y HA | Soporte para alta disponibilidad y replicación de estado. |
| Hooks personalizados | Se puede extender sin modificar el código fuente. |
| Soporte comercial | Ofrecido por ISC para entornos empresariales críticos. |

# Componentes principales

### 1. kea-dhcp4 / kea-dhcp6

Los servicios que asignan direcciones IPv4 o IPv6 respectivamente.

### 2. kea-ctrl-agent

Proporciona una API RESTful para gestionar Kea dinámicamente (reconfiguración, reservas, estadísticas, etc.).

### 3. kea-dhcp-ddns

Integración con DNS dinámico para actualizar registros DNS automáticamente.

### 4. Bases de datos backend

Permite almacenar leases (concesiones) y configuraciones en:

- MySQL
- PostgreSQL
- Cassandra

## Su uso para nuestra red

Para crear un servidor que de concesiones a nuestros clientes hemos decidido que vamos a utilizar “KEA” aquí nuestra red creada sera: *192.168.18.0*

Y para nuestro servicio de DNS hemos decidido utilizar bind9. Aquí hemos declarado la zona mast.smx2a de tipo “master”.

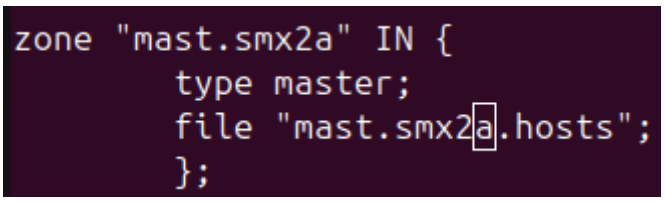



Figura 1.KEA (Declaración de zona)

Lo que tenemos que mirar es qué más poner en el bind, porque solo hemos declarado la zona y la hemos puesto en el archivo de /var/cache/bind/mast.smx2a.hosts. De momento creemos que está bien pero tenemos que hacer comprobaciones.

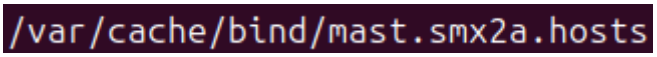



Figura 1.KEA (Configuración hosts)

# **HONEYPOT**:

Explicación:

Un **honeypot** es un sistema señuelo en la red diseñado para registrar y analizar ataques. Su propósito es detectar amenazas, alertar sobre posibles intrusiones y comprender mejor las técnicas de los atacantes.

Dado que no tiene un uso legítimo dentro de una organización, cualquier intento de acceso es sospechoso y facilita la identificación de actividades maliciosas. Engaña a los atacantes haciéndoles creer que están atacando un sistema real, pero en realidad operan en un entorno controlado.

Se pueden implementar dentro de la red (para detectar movimientos laterales) o fuera de ella (para identificar ataques externos). Además, varios honeypots pueden formar una **honeynet** para un monitoreo más amplio.

Los **honeypots** se clasifican según su nivel de interacción con los atacantes:

1. **De interacción alta**: Son servidores con servicios reales instalados, lo que permite recopilar información detallada sobre los ataques. Deben estar bien aislados para evitar que un atacante acceda a la red real. Son difíciles de detectar como trampas y permiten el uso de herramientas avanzadas, como la identificación del atacante mediante su fingerprint.

2. **De interacción media**: Emulan algunos servicios básicos (como HTTP o FTP) sin ser funcionales completamente. Generan menos información que los de alta interacción, pero son más fáciles de mantener y configuran respuestas programadas para los atacantes.

3. **De interacción baja**: Simulan solo protocolos de red básicos (TCP/IP, ICMP, NetBIOS), por lo que solo detectan escaneos de red. Son los más seguros, pero un atacante experimentado puede identificarlos fácilmente como honeypots.

# HONEYPOT DE INTERACCION ALTA (T- POT)

## Creación de nuestro honeypot → Proyecto T-Pot

### Que es el projecto T-POT?

El proyecto **T-Pot** es una plataforma de **honeypots** (trampas de ciberseguridad) de código abierto basada en **Docker**, desarrollada por Deutsche Telekom. Está diseñada para detectar y analizar ataques cibernéticos al simular sistemas vulnerables y atraer a atacantes reales.

### Tipos de instalación:

Hive - T-Pot Standard →Incluye todo el paquete para trampas, escaneos y más.

Sensor - T-Pot Sensor →Optimizado para una instalación distribuida sin Elasticsearch, Kibana y WebUI.

LLM - T- Pot LLM → Utiliza honeypots LLM basados en beelzebub y Galah.

Mini - T-Pot Mini →Instalación del T - Pot pero con menos cantidad, no incluye todo y prioriza el rendimiento para equipos menos potentes

Mobile - T-Pot Mobile →Incluye solo lo necesario para ejecutar T-Pot en dispositivos móviles.

Tarpit - T-Pot Tarpit → Honeypot que se encarga de enviar una sobrecarga de datos a los atacantes, bots y escáneres. También se puede usar como un negador de servicios (ddospot).

```
### Choose your T-Pot type:
### (H)ive    - T-Pot Standard / HIVE installation.
###             Includes also everything you need for a distributed setup with sensors.
### (S)ensor - T-Pot Sensor installation.
###             Optimized for a distributed installation, without WebUI, Elasticsearch and Kibana.
### (L)LM     - T-Pot LLM installation.
###             Uses LLM based honeypots Beelzebub & Galah.
###             Requires Ollama (recommended) or ChatGPT subscription.
### M(i)ni    - T-Pot Mini installation.
###             Run 30+ honeypots with just a couple of honeypot daemons.
### (M)obile - T-Pot Mobile installation.
###             Includes everything to run T-Pot Mobile (available separately).
### (T)arpit - T-Pot Tarpit installation.
###             Feed data endlessly to attackers, bots and scanners.
###             Also runs a Denial of Service Honeypot (ddospot).
```

Figura 1 Honeypot (Elección para la instalación del T-POT)

## Elección de instalacion de tipo en T-pot

Nosotros decidimos instalar el tipo Hive ya que viene el paquete completo y nos interesaba poder realizar el mayor número de pruebas con él.

**Configuracion:**

Usuario → t-potuser

Contraseña utilizada → 12345.Usuario

**Como conectarse al servidor por ssh :**

commando → ssh -p 64295 t-pot@192.168.12.203


Figura 2 Honeypot (Finalización de la instalación)

## Finalización de la instalación del honeypot y comprobación:

Pagina web del T-POT: https://192.168.12.203:64297/

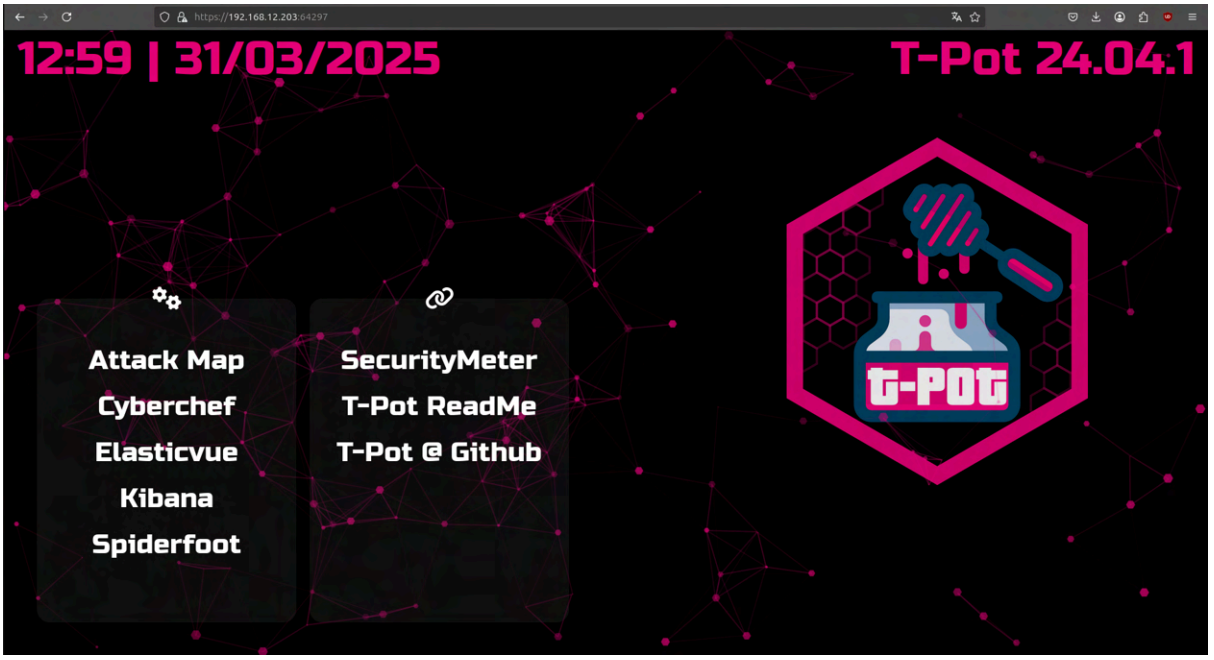



Figura 2 Honeypot (Interfaz principal t-pot)

## Problemas a la hora de instalar el T-Pot:

A la hora de instalar T-pot y este tener muchas dependencias y tener que abrir y ejecutar tantos programas acaba ocupando mucho espació, al principio pensé que no seria ningun problema pero al ejecutar los diferentes servicios y cargar las interfaces gráficas en formato web el servidor se quejaba de falta de memoria y no cargaba del todo la pagina, por eso mismo augmente la memoria de la máquina virtual para terminar de arreglarlo, sin embargo al cambiarlo pese a que ya no aparecia ese mensaje recurrente seguía sin cargar la interfaz gráfica. Cuando mire el espacio que ocupaba me fije que estaba utilizando el 98 % y al estar instalado en el mismo disco que el sistema operativo era propenso a explotar (dejar de funcionar).

Tuve bastantes problemas con el tema del espacio y la memoria y por eso tuve que informarme y al final encontré los requisitos del sistema y lei en foros de reddit entre otros, la misma situación que me había pasado a mi, al ser una máquina virtual y no tener tantos recursos no podemos permitirnos poner los requerimientos básicos pero debido a que funcionaba decidimos dejarlo así.

## Requisitos del sistema

Dependiendo de las imágenes de las distribuciones de Linux compatibles, del tipo de instalación (Hive o Sensor), y de si se instala en hardware real, en una máquina virtual u otros entornos, existen diferentes requisitos que deben cumplirse en cuanto a sistema operativo, RAM, almacenamiento y red para una instalación exitosa de T-Pot.

| Tipo de T-Pot | RAM | Almacenamiento | Descripción |
|---|---|---|---|
| Hive | 16 GB | 256GB SSD | Como regla general, a mayor cantidad de honeypots, sensores y datos, se requiere más RAM y almacenamiento. |
| Sensor | 8 GB | 128GB SSD | Dado que los registros de los honeypots se conservan (`~/tpotce/data`) durante 30 días, el almacenamiento necesario depende del volumen de ataques. |

## Lista de Honeypots en T-Pot

**SSH / Telnet / Shell**

**Beelzebub** – `22/tcp`
Simula un servidor SSH vulnerable que utiliza modelos de lenguaje para interactuar con atacantes, proporcionando una experiencia más realista e identificando comportamientos avanzados.

**Cowrie** – `22, 23/tcp`
Honeypot SSH y Telnet que simula un sistema Linux vulnerable. Registra comandos, credenciales y archivos utilizados por atacantes. Muy utilizado para análisis de tácticas y malware.

**Endlessh** – `22/tcp`
SSH honeypot de baja interacción que mantiene al atacante conectado indefinidamente sin responder, útil para ralentizar bots automatizados.

---

**Web / HTTP / HTTPS**

**Galah** – `80, 443, 8080, 8443/tcp`
Honeypot web de alta interacción que utiliza modelos de lenguaje para simular portales realistas y APIs expuestas. Ideal para analizar ataques web sofisticados.

**Go-pot** – `8080/tcp`
Honeypot HTTP ligero escrito en Go, diseñado para ser eficiente y fácil de desplegar, simula servicios web básicos.

**H0neytr4p** – `80, 443/tcp`
Simula servicios HTTP/HTTPS configurables con respuestas dinámicas. Útil para atraer atacantes web y analizar comportamiento automatizado.

**Snare (Tanner)** – `80/tcp`
Honeypot HTTP que simula servidores web con vulnerabilidades conocidas. Recoge intentos de explotación web.

**Wordpot** – `8090/tcp`
Simula sitios WordPress vulnerables para detectar ataques automáticos, como escaneos de plugins o inyecciones.

**Log4Pot** – `80, 443, 8080, 9200, 25565/tcp`
Detecta intentos de explotar la vulnerabilidad Log4Shell (Log4j). Simula servicios Java vulnerables comunes en entornos empresariales y de juegos.

---

**Servicios de red comunes**

**Heralding** – `21, 22, 23, 25, 80, 110, 143, 443, 993, 995, 1080, 5432, 5900/tcp`
Captura intentos de autenticación en múltiples servicios como FTP, IMAP, VNC, SMTP, y SSH. Útil para detectar robo de credenciales y ataques de fuerza bruta.

**qHoneypots** – `21, 22, 23, 25, 80, 110, 143, 389, 443, 445, 631, 1080, 1433, 1521, 3306, 3389, 5060, 5432, 5900, 6379, 6667, 8080, 9100, 9200, 11211/tcp`,
`53, 123, 161, 5060/udp`
Conjunto modular de honeypots que simulan una gran variedad de servicios comunes en una sola instancia, cubriendo protocolos de red, base de datos, impresión, y más.

**IPPHoney** – `631/tcp`
Simula servicios de impresión en red (IPP), frecuentemente expuestos en redes internas. Detecta exploraciones de servicios y acceso no autorizado a impresoras.

**Miniprint** – `9100/tcp`
Honeypot que emula impresoras de red usando protocolos como JetDirect. Permite detectar abuso o ataques a dispositivos de impresión.

---

**Industrial / IoT / SCADA**

**Conpot** – `80, 102, 502, 1025, 2404, 10001, 44818, 47808, 50100/tcp`, `161, 623/udp`
Simula dispositivos industriales como PLCs y controladores SCADA/ICS. Útil para analizar amenazas dirigidas a infraestructuras críticas.

**Medpot** – `2575/tcp`
Honeypot que emula dispositivos médicos conectados, ideal para pruebas de ciberseguridad en entornos de salud.

**Dicompot** – `11112/tcp`
Simula servicios DICOM utilizados por dispositivos de imágenes médicas. Permite identificar intentos de acceso a datos clínicos o de manipulación de imágenes.

---

**Dispositivos y servicios empresariales**

**CiscoASA** – `5000/udp, 8443/tcp`
Simula dispositivos de red como firewalls y VPN de Cisco ASA. Analiza intentos de explotación de dispositivos de seguridad de red.

**CitrixHoneypot** – `443/tcp`
Emula servidores Citrix para identificar ataques a escritorios remotos y gateways empresariales virtualizados.

**Elasticpot** – `9200/tcp`
Simula instancias de Elasticsearch expuestas públicamente. Detecta escaneos, exfiltración de datos y explotación de APIs.

---

**Correo y mensajería**

**Mailoney** – `25/tcp`
Honeypot SMTP diseñado para atraer y registrar intentos de envío de spam, phishing o abuso de servidores de correo.

---

**Bases de datos y almacenamiento**

**Redishoneypot** – `6379/tcp`
Simula instancias Redis mal configuradas o expuestas. Comúnmente explotadas para inyecciones, ransomware o minería de criptomonedas.

---

## VoIP / Telefonía

**SentryPeer** – `5060/tcp+udp`
Emula un sistema VoIP (SIP) para detectar escaneos y fraudes telefónicos como spoofing, robo de servicio y ataques de denegación.

---

**Otros**

**ADBHoney** – `5555/tcp`
Simula el Android Debug Bridge (ADB), que puede estar expuesto en dispositivos Android no asegurados. Usado frecuentemente por botnets.

**Ddospot** – `19, 53, 123, 1900/udp`
Detecta tráfico usado para ataques DDoS de amplificación mediante protocolos como NTP, DNS, SSDP.

**Dionaea** – `21, 42, 135, 443, 445, 1433, 1723, 1883, 3306, 8081/tcp`, `69/udp`
Honeypot multi-protocolo diseñado para capturar malware y estudiar su propagación a través de servicios comunes.

**Honeyaml** – `3000/tcp`
Honeypot configurable a través de archivos YAML que permite simular servicios personalizados de forma flexible.

# Kibana

## Logs y eventos (journal, syslog, etc.)

T-Pot recopila exhaustivamente registros de los honeypots y del sistema operativo. Esto incluye:

- **Logs detallados de honeypots individuales**
  Como Cowrie, Dionaea, Conpot, etc., que registran:
  - IP origen, timestamp
  - Comandos ejecutados
  - Credenciales usadas
  - Archivos subidos
  - Protocolo utilizado

- **Logs del sistema (journal)**

  T-Pot corre sobre Ubuntu y aprovecha `systemd`, por lo que los registros del sistema (journal) están disponibles y son recolectados si se configura.

- **Flujos de red (via Suricata y Zeek)**

  Captura tráfico con alta granularidad:

  - Detalles del flujo
  - Identificación de firmas (IDS)
  - Protocolo, payloads, conexiones

---

### Métricas y estadísticas

T-Pot usa un stack ELK (Elasticsearch, Logstash, Kibana) y a veces InfluxDB/Grafana (según la versión), para analizar datos recolectados. Las métricas incluyen:

- Número de ataques por puerto, hora, país de origen
- Tendencias en el tiempo
- Tipos de protocolos atacados
- Comportamientos comunes por honeypot

---

# SpiderFoot

SpiderFoot es una plataforma de recolección automatizada de inteligencia de fuentes abiertas **(OSINT)**. Diseñada para facilitar la exploración de superficies de ataque, esta herramienta es capaz de recolectar información pública sobre objetivos específicos como nombres de dominio, direcciones IP, correos electrónicos, nombres de usuario y más.

**¿Para qué se utiliza?**

SpiderFoot es ampliamente usada en procesos de reconocimiento pasivo y activo dentro de evaluaciones de seguridad. Su funcionalidad permite:

- Identificar relaciones entre diferentes elementos del entorno digital de un objetivo.
- Detectar posibles brechas de seguridad, credenciales filtradas o fugas de datos.
- Monitorear continuamente activos en busca de exposición pública no autorizada.

**¿Cómo funciona?**

SpiderFoot funciona mediante módulos configurables que consultan más de 200 fuentes públicas como Shodan, HaveIBeenPwned, VirusTotal, y bases de datos DNS. Puede ejecutarse en modo local o a través de una interfaz web, y también ofrece integración con otras plataformas mediante su API. (En nuestro caso lo utilizamos a través de una interfaz web en nuestro t-pot),

---

# CyberChef

CyberChef, desarrollado por el GCHQ (la agencia de inteligencia del Reino Unido), es una herramienta web interactiva para procesar, analizar y transformar datos. Es conocida como "el cuchillo suizo de los analistas de datos" por su versatilidad y facilidad de uso.

¿Para qué se utiliza?

Realizar operaciones de encriptación, compresión, codificación y análisis de datos de manera sencilla.

Además, es útil para descubrir secretos y decodificar datos que están ocultos, lo que puede ser de gran ayuda para investigaciones en ciberseguridad.

CyberChef permite:

- Convertir entre múltiples formatos (Base64, hexadecimal, binario, etc.).
- Cifrar y descifrar datos (AES, XOR, entre otros).
- Analizar archivos binarios o logs.
- Extraer hashes, direcciones IP, URLs, cadenas codificadas y más.

¿Cómo funciona?

CyberChef funciona mediante operaciones encadenadas a través de una interfaz tipo drag-and-drop (arrastrar y soltar). Los usuarios pueden aplicar múltiples transformaciones en secuencia sobre los datos a modo de capas, con resultados instantáneos, sin necesidad de conocimientos de programación.

---

## Elasticsearch

Elasticsearch es un motor de búsqueda y análisis distribuido basado en Apache Lucene (una biblioteca de búsqueda de texto completo escrita en Java). Se especializa en la indexación de grandes volúmenes de datos y permite realizar búsquedas complejas en tiempo real.

**¿Para qué se utiliza?**

Su uso es común en contextos como:

- Análisis de logs y monitoreo de sistemas.
- Construcción de motores de búsqueda internos.
- Visualización de métricas de seguridad o rendimiento.

**¿Cómo funciona?**

Elasticsearch organiza la información en documentos JSON, que se almacenan en índices. Gracias a su arquitectura distribuida, permite escalar fácilmente y mantener alta disponibilidad.

---

# Grafana, Prometheus y Loki

Hemos estado pensado una manera de monitorizar nuestros servicios y periféricos, ya que con la opción del T-POT que da esa forma de monitorización no nos pareció suficiente. Justo dio la casualidad de que con Victor Carceler, nuestro profesor de servicio de redes, hemos aprendido a ver cómo monitorizar nuestra red y nuestros ordenadores a través de un servicio llamado grafana. Este servicio es una servicio de monitorización gratuita y libre, grafana es muy amplio y con muchas funcionalidades.

Además de eso hemos utilizado otra herramienta llamada prometheus, esta herramienta lo que hace es monitorizar y recopilación de métricas. Aparte este se encarga de:

- Recolectar datos de métricas de servicios, aplicaciones, sistemas, etc., típicamente vía HTTP (los llamados endpoints /metrics).

- Almacenar esos datos en su base de datos interna basada en series temporales (time-series database).

- Ofrecer una API de consulta (PromQL) que permite buscar y filtrar métricas de forma potente.

En la administración moderna de sistemas y redes, es fundamental contar con herramientas que permitan **monitorizar el estado de los equipos y servicios**, así como gestionar los **logs** de forma eficiente. En este contexto, Grafana, Prometheus y Loki forman una **suite de herramientas muy potente** y ampliamente usada en entornos profesionales para la observabilidad.

---

## Grafana

**Grafana** es una plataforma de **visualización de datos** que se integra con Prometheus y muchas otras fuentes. Sirve para crear paneles gráficos interactivos donde se pueden ver en tiempo real los datos recopilados.

**Características principales:**

- Interfaz web muy intuitiva.

- Se pueden crear dashboards personalizados.

- Compatible con múltiples fuentes de datos: Prometheus, MySQL, InfluxDB, etc.

- Se puede usar para generar alertas visuales o notificaciones.

**Ejemplo de uso:**

Crear un panel donde se muestre el uso de CPU, RAM y red de todos los servidores de una red local.

---

## Prometheus

**Prometheus** es una herramienta de **monitorización y recolección de métricas**. Se encarga de recopilar información sobre el estado de los sistemas, como uso de CPU, memoria, tráfico de red, estado de servicios, etc.

**Características principales:**

- Recolecta métricas en formato **time-series** (series temporales).
- Utiliza un lenguaje propio llamado **PromQL** para hacer consultas.
- Funciona con un modelo de recolección *pull* (va a buscar los datos a los servicios).
- Almacena datos en su propia base de datos interna.
- Compatible con alertas mediante **Alertmanager**.

**Ejemplo de uso:**

Prometheus puede monitorizar servidores Linux y mostrar si hay sobrecarga de CPU o cuellos de botella en la RAM.

---

## Loki

**Loki** es una herramienta de **gestión centralizada de logs**, desarrollada por los creadores de Grafana. A diferencia de otras soluciones como ELK (Elasticsearch + Logstash + Kibana), Loki se centra en **ser fácil de integrar con Grafana** y consumir pocos recursos.

**Características principales:**

- Recoge logs y los organiza por etiquetas, como el nombre del host o del contenedor.
- Funciona de forma similar a Prometheus, pero para logs en lugar de métricas.
- Muy eficiente en almacenamiento.
- Se puede consultar y visualizar directamente desde **Grafana**.

**Ejemplo de uso:**

Centralizar los logs de todos los servidores para poder buscar errores o mensajes concretos sin tener que ir equipo por equipo.

---

## Cómo se relacionan entre sí

Estas tres herramientas pueden integrarse para formar un sistema completo de monitorización:

- **Prometheus** recolecta y guarda métricas.
- **Grafana** muestra las métricas y los logs en dashboards.
- **Loki** centraliza los logs para que también se puedan visualizar desde Grafana.

  Esta integración permite tener **una visión completa** del estado de una infraestructura de red, tanto a nivel de rendimiento como de errores o advertencias.

## Ventajas para técnicos de sistemas

- Mejora el control sobre los servidores y servicios.
- Ayuda a detectar fallos o cuellos de botella antes de que afecten al usuario.
- Ahorra tiempo en la búsqueda de errores (gracias a Loki).
- Es software libre y gratuito.
- Facilita el trabajo en equipos IT profesionales.

## Conclusión sobre su uso

Grafana, Prometheus y Loki forman un conjunto de herramientas muy completo para tareas de **monitorización y observabilidad**, cada vez más utilizado en entornos reales. Para un estudiante de SMX, aprender a utilizarlas aporta una ventaja competitiva, ya que son herramientas demandadas en el mundo laboral y aplicables tanto en pequeñas como grandes infraestructuras.

# Ansible

Ansible es una herramienta de automatización de código abierto que permite gestionar configuraciones, desplegar aplicaciones y orquestar tareas de administración en múltiples equipos desde un único punto. Es especialmente útil en entornos con muchos servidores, ya que reduce el tiempo de trabajo manual y asegura que las configuraciones sean consistentes.

---

## ¿Cómo funciona Ansible?

Ansible se basa en una arquitectura sin agentes (*agentless*), lo que significa que no necesita instalar software adicional en los equipos que gestiona. Se conecta mediante **SSH** (para sistemas Linux/Unix) o **WinRM** (para sistemas Windows) y ejecuta tareas definidas en archivos escritos en **YAML**, llamados *Playbooks*.

---

## Componentes principales

- **Nodo de control**: Es el equipo desde el que se ejecutan los comandos de Ansible.
- **Nodos gestionados**: Son los equipos o servidores sobre los que se aplica la automatización.
- **Inventario**: Es un archivo que contiene una lista de los nodos gestionados, agrupados según su función o características.
- **Playbooks**: Archivos YAML donde se definen las tareas o configuraciones que se desean aplicar.
- **Módulos**: Funciones predefinidas que permiten realizar acciones concretas (instalar software, copiar archivos, gestionar servicios, etc.).
- **Roles**: Organización modular de los playbooks que facilita la reutilización y mantenimiento del código.

---

### Funcionalidades de Ansible

- **Automatización de configuración**: Permite configurar de forma automática el software y los servicios en uno o varios equipos.
- **Despliegue de aplicaciones**: Facilita la instalación y puesta en marcha de aplicaciones en diferentes entornos.
- **Orquestación de sistemas**: Coordina múltiples tareas y servicios que deben ejecutarse en distintos servidores.
- **Gestión de actualizaciones**: Se pueden aplicar parches y actualizaciones de forma centralizada.
- **Uso de Ansible Vault**: Permite encriptar información sensible, como contraseñas o claves de acceso, para mayor seguridad.

### Ventajas de utilizar Ansible

- **Fácil de aprender y usar** gracias a su sintaxis clara en YAML.
- **No requiere agentes**, lo que facilita su implementación.
- **Compatible con múltiples sistemas operativos**.
- **Altamente escalable**: se puede usar tanto en pequeños entornos como en grandes infraestructuras.
- **Amplia comunidad y soporte**: al ser una herramienta muy utilizada, dispone de mucha documentación y recursos.

---

### Conclusión sobre su uso

Ansible es una herramienta muy útil en el ámbito de la administración de sistemas, especialmente en entornos donde se requiere rapidez, eficiencia y coherencia en las configuraciones. Su uso en entornos reales está muy extendido, por lo que aprender a manejarla correctamente es una ventaja importante para cualquier técnico de sistemas, especialmente en el contexto de estudios como SMX.

# Precios

Tabla de precios aproximados para los servicios de nuestra empresa →

| Servicio | Precio Instalación Particulares | Precio Instalación Empresas | Descripción breve |
|---|---|---|---|
| **DHCP (Kea)** | 100 € | 250 € | Asignación segura y organizada de IPs en red. |
| **DNS (Bind9)** | 120 € | 300 € | Gestión de dominios para entornos personales y empresariales. |
| **Grafana** | 150 € | 400 € | Monitorización de red con paneles gráficos avanzados. |
| **Ansible** | 180 € | 500 € | Automatización y despliegue rápido de configuraciones. |
| **Mantenimiento** | 80 €/hora | 100 €/hora | Soporte técnico para hardware y software. |
| **Honeypot** | 130 € | 350 € | Seguridad mediante detección de amenazas en la red. |

Lo que se puede ver en esta tabla es una valor aproximado de lo que puede costar dependiendo el tamaño de la red, infraestructura existente o requerimientos especiales, nosotros no tenemos costes fijos ya que lo que hacemos son costes personalizados por cada cliente dependiendo sus necesidades.

También se podrían ofrecer paquetes o descuentos por la contratación de varios servicios juntos.

# Resultados / Objetivos cumplidos:

En este apartado tenemos imágenes que muestran nuestros servicios en funcionamiento:

## T-pot y sus herramientas:

-Kibana

-Spiderfoot

-Ciberchef

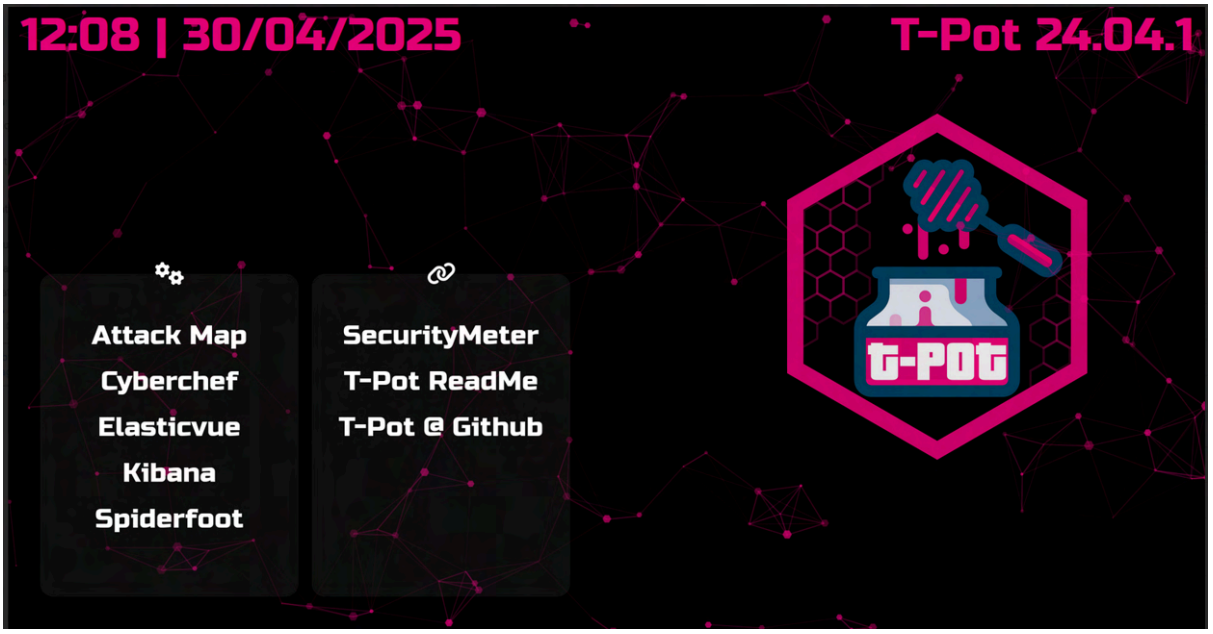



Figura 2 Honeypot (Interfaz principal t-pot)


Figura 3 Honeypot (Dashboard Kibana)

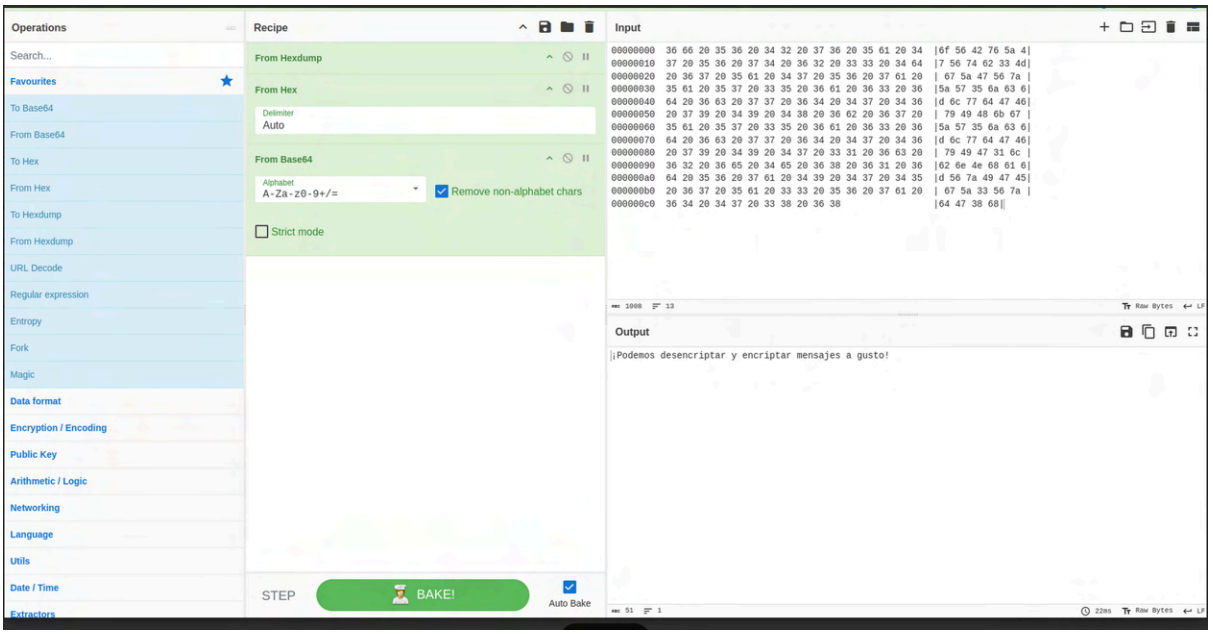



Figura 4 Honeypot (Interfaz Cyber chef)

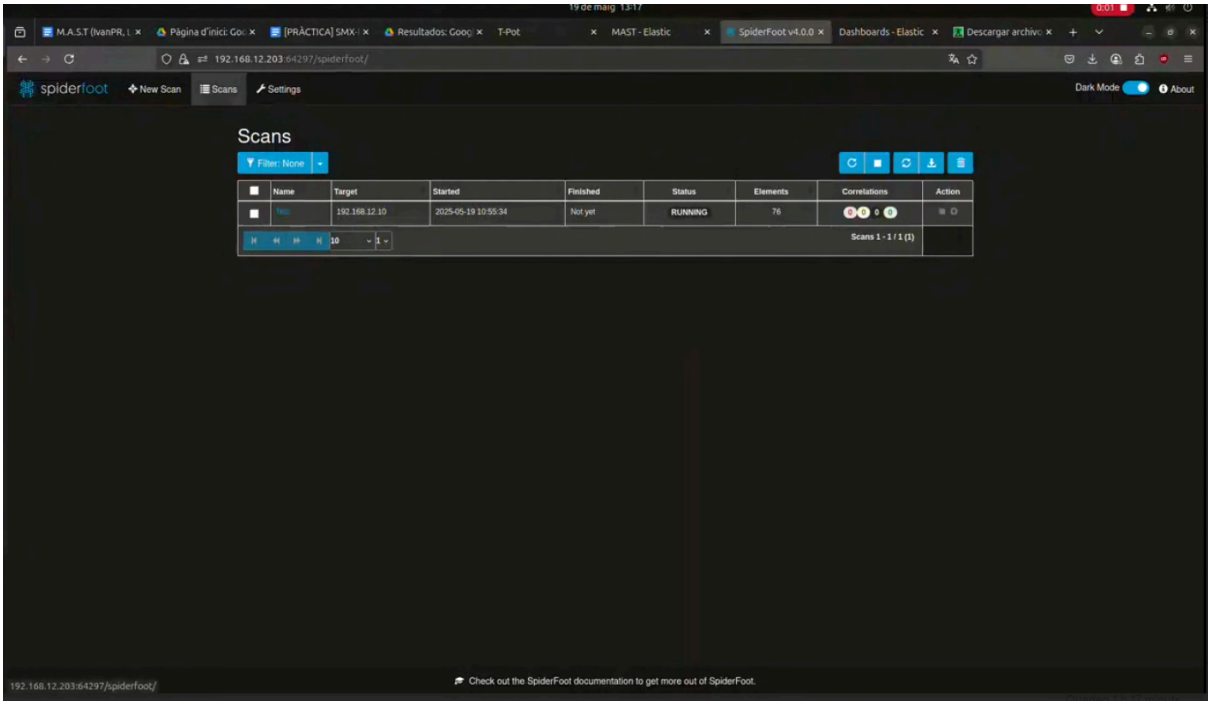



Figura 5 Honeypot (Interfaz de Spiderfoot)

# Grafana, Loki y Prometheus

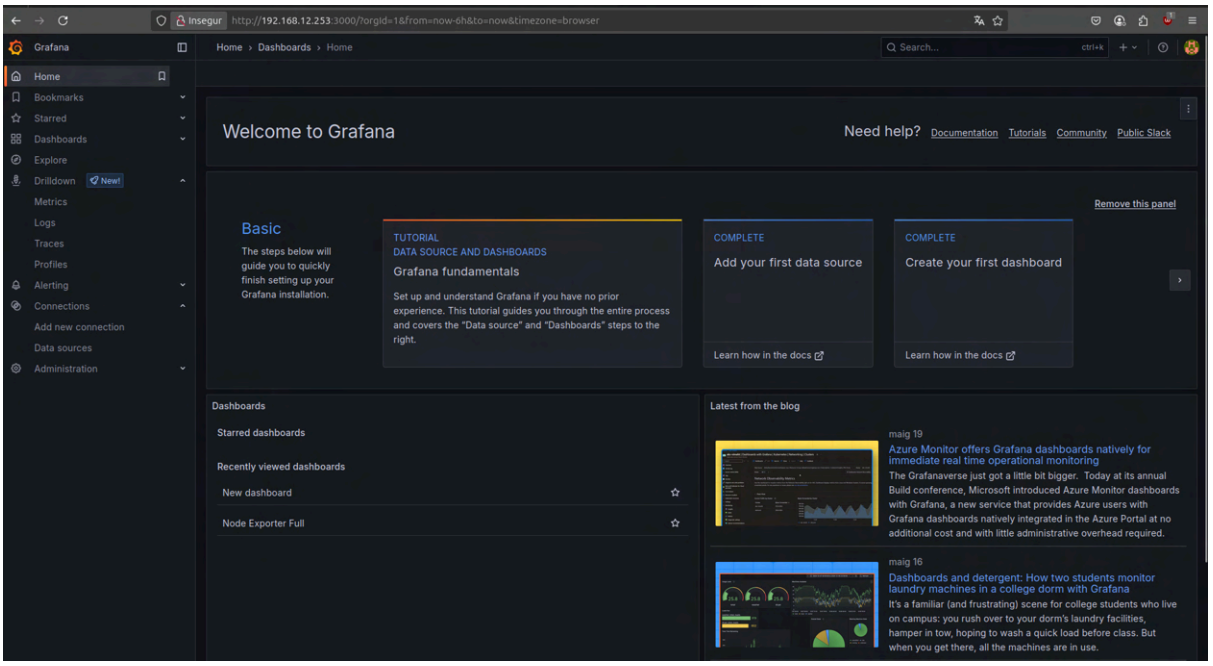



Figura 1 Grafana (Interfaz Grafana)

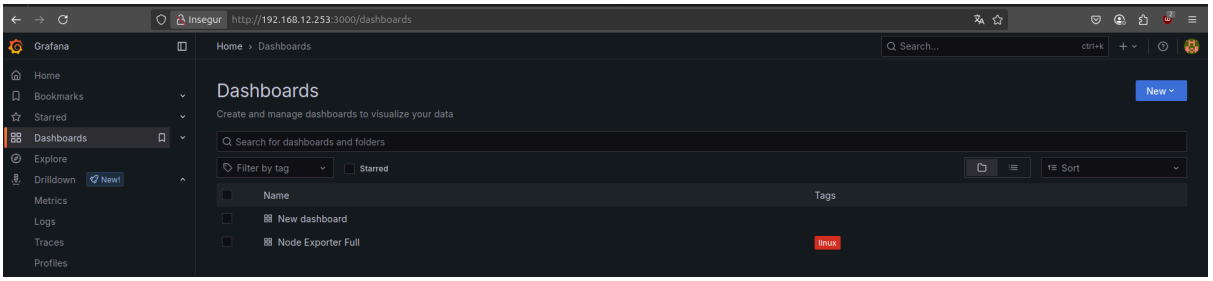



Figura 2 Grafana (Interfaz creación de dashboard)


Figura 3 Grafana (Dashboard grafana)

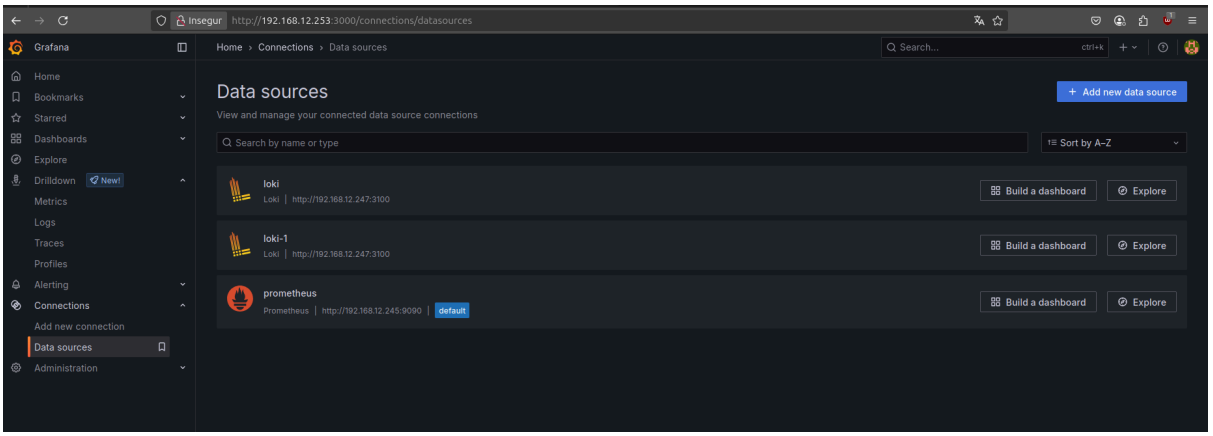



Figura 4 Grafana (Data sources utilizados)

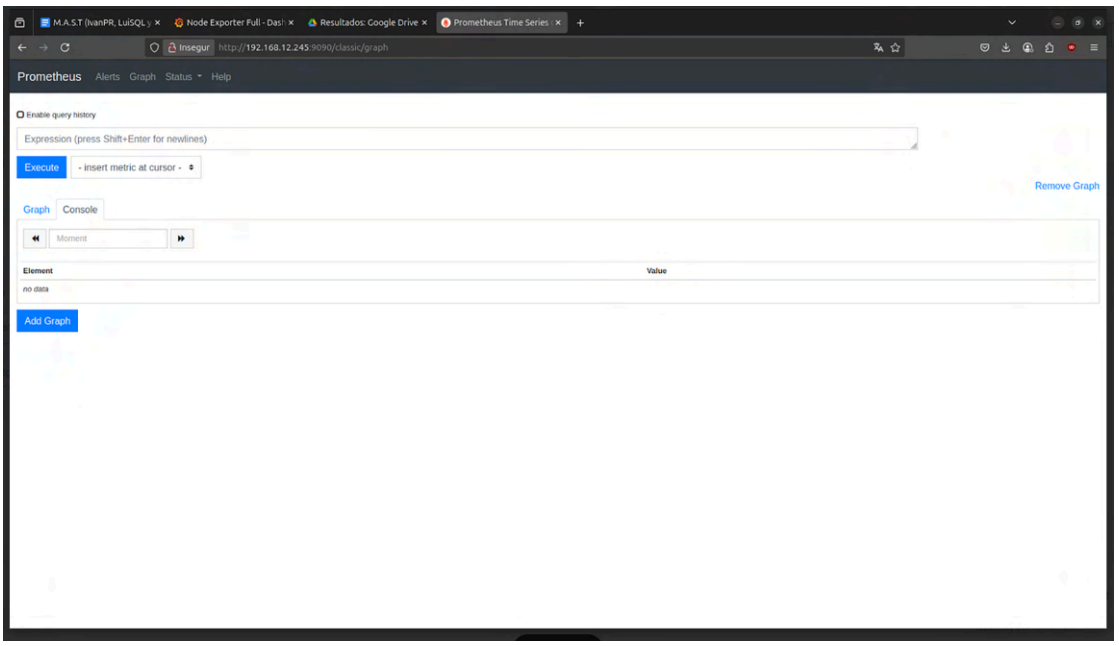



Figura 1 Prometheus (Interfaz Prometheus)

```
# Sample config for Prometheus.

global:
  scrape_interval:     15s # Set the scrape interval to every 15 seconds. Default
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is ever
  # scrape_timeout is set to the global default (10s).

  # Attach these labels to any time series or alerts when communicating with
  # external systems (federation, remote storage, Alertmanager).
  external_labels:
      monitor: 'example'

# Alertmanager configuration
alerting:
  alertmanagers:
  - static_configs:
    - targets: ['localhost:9093']

# Load rules once and periodically evaluate them according to the global 'evaluat
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped f
  - job_name: 'equipos'

    # Override the global default and scrape targets from this job every 5 seconc
    scrape_interval: 5s
    scrape_timeout: 5s

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ['192.168.12.245:9090']

  - job_name: node
    # If prometheus-node-exporter is installed, grab stats about the local
    # machine by default.
    static_configs:
      - targets: ['localhost:9100','192.168.18.201:9100','192.168.18.202:9100']
~
~
~
~
~
"/etc/prometheus/prometheus.yml" [Sólo lectura] 44L, 1475B
```

Figura 2 Prometheus (Configuración Prometheus)

# Ansible

```
usuario@mast-201:~$ cat /etc/hostname
mast-201
usuario@mast-201:~$
```

Figura 1 Ansible (Ansible en pc-1)

```
root@ansible:/home/usuario# ansible-playbook -i hosts playbook.yml

PLAY [equipos] ***************************************************************************************************************************

TASK [Gathering Facts] *******************************************************************************************************************
ok: [192.168.18.204]
ok: [192.168.18.200]
ok: [192.168.18.203]
ok: [192.168.18.202]
ok: [192.168.18.205]
ok: [192.168.18.201]

TASK [Configura hostname si es diferente al deseado] *************************************************************************************
skipping: [192.168.18.200]
skipping: [192.168.18.201]
skipping: [192.168.18.202]
skipping: [192.168.18.203]
skipping: [192.168.18.204]
skipping: [192.168.18.205]

TASK [Actualiza /etc/hosts con el nuevo hostname] ****************************************************************************************
ok: [192.168.18.204]
ok: [192.168.18.202]
ok: [192.168.18.203]
ok: [192.168.18.200]
ok: [192.168.18.201]
ok: [192.168.18.205]
```

Figura 2 Ansible  (Playbook de ansible en funcionamiento)

```
TASK [Hace apt update] ***********************************************************
changed: [192.168.18.204]
changed: [192.168.18.202]
changed: [192.168.18.201]
changed: [192.168.18.200]
changed: [192.168.18.203]
changed: [192.168.18.205]

TASK [Reinicia el sistema si el hostname fue cambiado] ************************
skipping: [192.168.18.200]
skipping: [192.168.18.202]
skipping: [192.168.18.201]
skipping: [192.168.18.204]
skipping: [192.168.18.203]
skipping: [192.168.18.205]

PLAY RECAP *********************************************************************
192.168.18.200             : ok=3    changed=1    unreachable=0    failed=0    skipped=2
 rescued=0    ignored=0
192.168.18.201             : ok=3    changed=1    unreachable=0    failed=0    skipped=2
 rescued=0    ignored=0
192.168.18.202             : ok=3    changed=1    unreachable=0    failed=0    skipped=2
 rescued=0    ignored=0
192.168.18.203             : ok=3    changed=1    unreachable=0    failed=0    skipped=2
 rescued=0    ignored=0
192.168.18.204             : ok=3    changed=1    unreachable=0    failed=0    skipped=2
 rescued=0    ignored=0
192.168.18.205             : ok=3    changed=1    unreachable=0    failed=0    skipped=2
 rescued=0    ignored=0
```

Figura 3 Ansible (Playbook finalizado)

# Video promocional en inglés

Luis Rafa Ivan

- Okay, we need to explain to clients how our honeypot, DHCP, and DNS services are crucial to their security.

- You're right. Sometimes they don't realize the importance of these components. Let's start with honeypots.

- A honeypot is like setting a delicious trap for hackers. It's a decoy system designed to attract them so we can study their methods without affecting our real systems.

- And let's not forget, of course, DHCP. It's sometimes underestimated, but a well-configured DHCP server can prevent spoofing attacks and keep your network organized.

- Absolutely. We offer comprehensive DHCP management, from configuration to troubleshooting, etc.

- And of course, DNS. A secure DNS is the foundation of reliable web browsing. We protect your DNS servers against DDoS, DoS, and other risks.

- That's right. We implement robust security measures and offer fast, reliable name resolution so your website and services are always available.

- In short, with our honeypot, DHCP, and DNS services, we offer a comprehensive layer of security that protects your infrastructure from multiple angles.

- And all with expert technical support and the peace of mind of knowing your systems are in good hands with us.

# CUADERNO DE BITÁCORA

Esta tabla muestra distintos colores porque cada color está asociado a un mes específico:

| Mes | Piedra de nacimiento | Color asociado | Fuente |
|---|---|---|---|
| Enero | Granate | Rojo oscuro | The Old Farmer's Almanac |
| Febrero | Amatista | Púrpura | The Old Farmer's Almanac |
| Marzo | Aguamarina | Azul claro | The Old Farmer's Almanac |
| Abril | Diamante | Transparente o blanco | The Old Farmer's Almanac |
| Mayo | Esmeralda | Verde | The Old Farmer's Almanac |
| Junio | Perla, Alejandrita | Blanco (perla), Verde (alejandrita) | The Old Farmer's Almanac |
| Julio | Rubí | Rojo | The Old Farmer's Almanac |
| Agosto | Peridoto | Verde lima | The Old Farmer's Almanac |
| Septiembre | Zafiro | Azul profundo | The Old Farmer's Almanac |
| Octubre | Ópalo, Turmalina | Multicolor (ópalo), Rosa (turmalina) | The Old Farmer's Almanac |
| Noviembre | Topacio, Citrino | Amarillo (topacio), Naranja (citrino) | The Old Farmer's Almanac |
| Diciembre | Turquesa, Tanzanita, Circón azul | Azul celeste (turquesa), Azul violáceo (tanzanita), Azul (circón) | The Old Farmer's Almanac |

| Fecha | Trabajadores | Tareas realizadas |
|---|---|---|
| 09/09/24 | Rafa, Ivan | Buscar y compartir ideas sobre el proyecto a realizar + Documentación (Lluvia de ideas). |
| 11/09/24 | Rafa, Ivan | Decidimos decantarnos por un proyecto enfocado en la ciberseguridad pero seguimos buscando información y barajando ideas. |
| 12/09/24 | Rafa, Ivan | Salió la idea del ratón “hacker” (una coña entre amigos) y empezamos con bocetos para el logo. |
| 16/09/24 | Rafa, Ivan | Hicimos la misión, visión y valores sobre nuestra empresa. |
| 18/09/24 | Rafa, Ivan | Se definió la misión para orientar el propósito del proyecto, que es proveer servicios de monitoreo y protección cibernética mediante el uso de honeypots. La visión busca posicionarnos como una referencia de seguridad en el ámbito educativo y empresarial. |

| | | |
|---|---|---|
| 19/09/24 | Rafa, Ivan | Después de un tiempo con los bocetos para el logo decidimos probar la IA (inteligencia artificial) que crea imágenes para ver qué podía salir de ello y al final nos quedamos con el que hizo la IA. |
| 23/09/24 | Rafa, Ivan | Al comentarle nuestras ideas a nuestros profesores nos dijeron que no era buena idea por la complejidad que conllevaba hacerlo así que decidimos descartar la idea y tuvimos que darle otro enfoque. |
| 25/09/24 | Rafa, Ivan | Buscamos e indagamos proyectos sencillos que podíamos hacer con nuestros estudios sobre ciberseguridad e hicimos otra lluvia de ideas. |
| 26/09/24 | Rafa, Ivan | Finalmente nos decantamos por dos ideas principales que no eran tan complejas y que podíamos realizar como es el caso de crear un honeypot y un usb autoinyectable que ejecute un script dañino. |
| 30/09/24 | Rafa, Ivan | Recopilamos información sobre el concepto de honeypots, su funcionamiento, y cómo se pueden implementar y |

| | | |
|---|---|---|
| | | personalizar para hacerlos más efectivos en la detección de ataques. |
| 2/10/24 | Rafa, Ivan | Buscamos información sobre cuales podíamos instalar y personalizar pero que fueran de código abierto + comienzo del diagrama de Gantt. |
| 3/10/24 | Rafa, Ivan | Vimos cómo funcionan los honeypots y encontramos T-Pot como herramienta clave. |
| 7/10/24 | Rafa, Ivan | Exploramos los tipos de ataques más comunes en la actualidad, como phishing, ransomware y ataques DDoS, y cómo pueden ser detectados utilizando honeypots. |
| 9/10/24 | Rafa, Ivan | Avanzamos contenido del plan de empresa como la presentación de los promotores y una tabla de nuestras virtudes y defectos. |
| 10/10/24 | Rafa, Ivan | Realizamos el diagrama de gantt y un gráfico de análisis de riesgos. |
| 14/10/24 | Rafa, Ivan | Decidimos la estrategia que utilizaremos para organizarnos, la metodología de trabajo y el tiempo estimado que dedicamos a cada |

| | | |
|---|---|---|
| | | tarea. |
| 16/10/24 | Rafa, Ivan | Definimos los objetivos, metas y submetas a realizar para poder organizarnos correctamente |
| 17/10/24 | Rafa, Ivan | Documentación, sobre todo lo que íbamos encontrando, diseño que queríamos hacer de la página web y que mini proyectos haríamos para cubrir todas las materias. |
| 21/10/24 | Rafa, Ivan | Definimos finalmente los mini proyectos que haríamos para cubrir las asignaturas y nos pusimos a ello. |
| 23/10/24 | Rafa, Ivan | Creación de una red, servidor DHCP y DNS, que sirviera como router y sirviera como puente a la red del centro. |
| 24/10/24 | Rafa, Ivan | Solución de problemas al realizar tareas en nuestra red. |
| 28/10/24 | Rafa, Ivan | Documentación y búsqueda de información mediante videos y páginas para crear e instalar el honeypot. |
| 30/10/24 | Rafa, Ivan | Avanzamos en el plan de empresa creando el apartado de marketing |

| | | |
|---|---|---|
| | | y las estrategias que utilizaremos para nuestra empresa ficticia. |
| 31/10/24 | Rafa, Ivan | Avanzamos en plan de empresa realizando documentación y cambios en algunos apartados ya realizados. |
| 04/11/24 | Rafa, Ivan | Recopilación de información sobre t-pot y su uso. |
| 06/11/24 | Rafa, Ivan | Recopilación de información sobre cómo crear el usb autoinyectable dañino. |
| 07/11/24 | Rafa, Ivan | Descarte parcial de crear el usb dañino + documentación. |
| 11/11/24 | Rafa, Ivan | Documentación sobre plan de empresa y avanzado en la memoria. |
| 13/11/24 | Rafa, Ivan, Luis | Luis se unió al grupo y reorganizamos tareas para poder avanzar con más facilidad. |
| 14/11/24 | Rafa, Ivan, Luis | Al unirse Luis cuando llevábamos ya avanzados en proyecto tuvimos que explicarle nuestro progreso como nos estábamos dividiendo las tareas informarle de que cosas estábamos haciendo y |

| | | |
|---|---|---|
| | | posteriormente organizando las tareas conforme todos hicieramos una parte. |
| 18/11/24 | Rafa, Ivan, Luis | Con Luis en el equipo empezamos las tareas de documentación para documentar todo de forma más organizada conforme pudiera entregarse, asi seria mas facil de encontrar las cosas y ya seguir con un buen ritmo a la vez que limpio. |
| 20/11/24 | Rafa, Ivan, Luis | Otro día organizando nuestra memoria y documentando con más fuentes y corrigiendo faltas. |
| 21/11/24 | Rafa, Ivan, Luis | Nos metimos de lleno a investigar los diferentes tipos de honeypots y sus aplicaciones. Analizamos ejemplos de bajo y alto nivel de interacción, y cómo se utilizan en escenarios reales. Esto nos dio una idea más clara sobre qué tipo de honeypot sería más útil según el objetivo del proyecto. |
| 25/11/24 | Rafa, Ivan, Luis | Ya tenemos una base sólida de información recopilada sobre T-Pot, honeypots, ataques comunes y empresas del sector. Este trabajo nos está permitiendo ver el panorama |

| | | |
|---|---|---|
| | | completo y nos da una guía clara para los siguientes pasos. El enfoque ahora va hacia análisis más detallados y posibles pruebas de laboratorio. |
| 27/11/24 | Rafa, Ivan, Luis | Comienzo de la instalación del servidor que hará de honeypot con t-pot, creación de un usuario en red hat (fedora) e inicio de la instalación de wordpress para la página. |
| 28/11/24 | Rafa, Ivan, Luis | Fallos de la instalación del honeypot, documentación y avanzado en el uso de Elementor. |
| 02/12/24 | Rafa, Ivan, Luis | Consultamos con los profesores una posible solución a nuestro problema y planteamos una serie de soluciones que fuimos probando una a una. |
| 04/12/24 | Rafa, Ivan, Luis | Conseguimos solucionar la falla del honeypot y así que funcione correctamente + documentación del proceso. |
| 05/12/24 | Rafa, Ivan, Luis | Hicimos pruebas con el honeypot ya instalado, haciendo un escaneo del honeypot con nmap para ver si en los dashboards se |

| | | |
|---|---|---|
| | | mostraba como alguien había intentado escanearlo y documentamos el resultado. |
| 09/12/24 | Rafa, Ivan, Luis | Avanzamos en la creación del usuario de Red Hat y documentamos diferencias entre Ubuntu y Red Hat. |
| 11/12/24 | Rafa, Ivan, Luis | Documentación sobre Red Hat y su uso para nuestros fines. |
| 12/12/24 | Rafa, Ivan, Luis | Documentación y búsqueda de información. |
| 16/12/24 | Rafa, Ivan, Luis | Pruebas con el honeypot y la página web. |
| 18/12/24 | Rafa, Ivan, Luis | Documentación + organización de lo descubierto hasta hoy. |
| 19/12/24 | Rafa, Ivan, Luis | Documentación + corrección de faltas de la memoria. |
| 23/12/24 | Rafa, Ivan, Luis | Seguimos documentando y buscando información con tal de conseguir firmeza en nuestros apuntes. |
| **25/12/24** | **Vacaciones de Navidad (Descansito)** | |

| | | |
|---|---|---|
| **26/12/24** | **Vacaciones de Navidad (Descansito)** | |
| **30/12/24** | **Vacaciones de Navidad (Descansito)** | |
| **01/01/25** | **Vacaciones de Navidad (Descansito)** | |
| **02/01/25** | **Vacaciones de Navidad (Descansito)** | |
| **06/01/25** | **Vacaciones de Navidad (Descansito)** | |
| 08/01/25 | Rafa, Ivan, Luis | Nos hemos estado dedicando a investigar y recopilar información. |
| 09/01/25 | Rafa, Ivan, Luis | Tuvimos un error en el que por accidente un integrante del equipo restauro una versión del documento anterior y tuvimos que volver a organizar algunas cosas que perdimos |
| 13/01/25 | Rafa, Ivan, Luis | Seguimos con la restauración de la documentación perdida. |
| 15/01/25 | Rafa, Ivan, Luis | Conseguimos restaurar gran parte de lo perdido gracias a que ya sabíamos los sitios utilizados para la búsqueda de información. |
| 16/01/25 | Rafa, Ivan, Luis | Después de este problema seguimos con la investigación y |

| | | |
|---|---|---|
| | | recopilar información. |
| 20/01/25 | Rafa, Ivan, Luis | Nos dedicamos a hacer la parte de RRHH y acabar la parte de marketing |
| 22/01/25 | Rafa, Ivan, Luis | Fuimos implementando contenido y mejorando la página Web |
| 23/01/25 | Rafa, Ivan, Luis | Investigamos y utilizamos Ansible para facilitar nuestras tareas de forma rápida y eficiente. |
| 27/01/25 | Rafa, Ivan, Luis | Seguimos investigando y avanzando en general durante los siguientes días. |
| 29/01/25 | Rafa, Ivan, Luis | x2 Seguimos investigando y avanzando en general durante los siguientes días. |
| 30/01/25 | Rafa, Ivan, Luis | x3 Seguimos investigando y avanzando en general durante los siguientes días. |
| 03/02/25 | Rafa, Ivan, Luis | Acabamos nuestro RRHH y empezamos con el Plan financiero/inversiones. |
| 05/02/25 | Rafa, Ivan, Luis | Investigamos y utilizamos Grafana + documentación sobre su funcionalidad |

| | | |
|---|---|---|
| 06/02/25 | Rafa, Ivan, Luis | Seguimos avanzado la memoria. |
| 10/02/25 | Rafa, Ivan, Luis | Fuimos realizando pruebas / ataques hacia el honeypot + documentación. |
| 12/02/25 | Rafa, Ivan, Luis | Seguimos avanzado el plan de empresa + pruebas / ataques hacia el honeypot. |
| 13/02/25 | Rafa, Ivan, Luis | Documentación de todas las observaciones del dia anterior. |
| 17/02/25 | Rafa, Ivan, Luis | Volvimos con Grafana para crear nuestros dashboards sobre el proyecto + imágenes + documentación. |
| 19/02/25 | Rafa, Ivan, Luis | Volvimos a hacer pruebas con la Página Web. |
| 20/02/25 | Rafa, Ivan, Luis | Revisamos Grafana para asegurarnos de que funcione correctamente. |
| 24/02/25 | Rafa, Ivan, Luis | Finalizamos con el Plan financiero/inversiones (Balance de situación etc). |
| 26/02/25 | Rafa, Ivan, Luis | Empezamos a crear nuestra cuenta de Instagram del proyecto. |
| 27/02/25 | Rafa, Ivan, Luis | Finalizamos con la |

| | | cuenta de Instagram |
|---|---|---|
| 03/03/25 | Rafa, Ivan, Luis | Comenzamos a realizar fotografías y videos del funcionamiento de todos nuestros servicios, como plan b en caso de que el día de la presentación no funcionase correctamente. |
| 05/03/25 | Rafa, Ivan, Luis | Fuimos subiendo las Máquinas virtuales a nuestro drive para después compartirla con los profesores. |
| 06/03/25 | Rafa, Ivan, Luis | Añadimos las fotografías realizadas anteriormente a la memoria en un nuevo apartado de resultados |
| 10/03/25 | Rafa, Ivan, Luis | Investigación de las herramientas que contiene T-pot mediante (videos, github, foros, etcetera…..) |
| 12/03/25 | Rafa, Ivan, Luis | Instalación completa de T-pot + documentación. |
| 13/03/25 | Rafa, Ivan, Luis | Corrección de faltas de ortografía entre otras cosas de la memoria. |
| 17/03/25 | Rafa, Ivan, Luis | Terminando de completar la memoria haciendo arreglos y añadiendo nuevos apartados. |

| | | |
|---|---|---|
| 19/03/25 | Rafa, Ivan, Luis | Terminamos con el apartado de agradecimientos. |
| 20/03/25 | Rafa, Ivan, Luis | Terminamos con el apartado de conclusiones y cumplimentación de objetivos. |
| 24/03/25 | Rafa, Ivan, Luis | Terminamos con el apartado de posibles mejoras de nuestro proyecto. |
| 26/03/25 | Rafa, Ivan, Luis | Corrección de algunas partes de la memoria por sugerencia de Federico Fedez. |
| 27/03/25 | Rafa, Ivan, Luis | Finalizamos con la Página Web |
| 31/03/25 | Rafa, Ivan, Luis | Empezamos a hacer un video promocional para el proyecto. |
| 02/04/25 | Rafa, Ivan, Luis | Comenzamos con la presentación en Canva (plan de empresa). |
| 07/04/25 | Rafa, Ivan, Luis | Empezamos con Canva el organigrama del proyecto. |
| 08/04/25 | Rafa, Ivan, Luis | Consultamos con nuestro profesor (Jose Camuñez) si nuestro proyecto estaba bien encaminado y correcto. |
| 10/04/25 | Rafa, Ivan, Luis | Organización de las tareas a realizar en |

| | | |
|---|---|---|
| | | semana santa y distribución equitativa. |
| 14/04/25 | **Vacaciones de Semana Santa (Descansito)** | |
| 16/04/25 | **Vacaciones de Semana Santa (Descansito)** | |
| 17/04/25 | **Vacaciones de Semana Santa (Descansito)** | |
| 21/04/25 | **Vacaciones de Semana Santa (Descansito)** | |
| 23/04/25 | Rafa, Ivan, Luis | Cyberchef info<br>Spiderfoot info<br>Elasticsearch info |
| 24/04/25 | Rafa, Ivan, Luis | Pruebas con Cyber Chef. + documentación sobre su uso. |
| 28/04/25 | Rafa, Ivan, Luis | No se pudo avanzar mucho debido al apagón en España. |
| 30/04/25 | Rafa, Ivan, Luis | Investigamos sobre Elasticsearch + documentación sobre su uso. (Vimos que no podríamos utilizarlo) |
| 05/05/25 | Rafa, Ivan, Luis | Finalización del Video Promocional. |
| 07/05/25 | Rafa, Ivan, Luis | Finalización con la presentación en Canva |

| | | |
|---|---|---|
| | | (plan de empresa) + Finalización del organigrama del proyecto. |
| 12/05/25 | Rafa, Ivan, Luis | Finalización del Diagrama de Gantt. |
| 14/05/25 | Rafa, Ivan, Luis | Hicimos uso de Spiderfoot + documentación sobre su uso. |
| 19/05/25 | Rafa, Ivan, Luis | Corregimos la memorias (errores ortográficos, espacios en blanco etc). |
| 21/05/25 | Rafa, Ivan, Luis | Corregimos la memorias (errores ortográficos, espacios en blanco etc). |
| 26/05/25 | Rafa, Ivan, Luis | Ensayamos para el día de la exposición. |
| 28/05/25 | Rafa, Ivan, Luis | Ensayamos para el día de la exposición. |

# Conclusión Final del proyecto

El desarrollo del proyecto **M.A.S.T** ha representado una experiencia práctica integral que ha permitido aplicar de forma real los conocimientos adquiridos a lo largo del Ciclo Formativo de Grado Medio en Sistemas Microinformáticos y Redes.

Durante su ejecución, creemos haber demostrado capacidad para planificar, investigar, implementar y documentar soluciones informáticas orientadas a la **ciberseguridad** y a la administración de redes, utilizando herramientas y metodologías actuales del sector.

Entre los logros más destacados del proyecto se encuentran:

- La **instalación y configuración completa de un honeypot de alta interacción (T-Pot)**, con una gran variedad de servicios simulados y herramientas de análisis como Kibana y Spiderfoot.

- La **creación de una página web profesional** mediante WordPress, orientada a la presentación de servicios informáticos en un contexto empresarial simulado.

- La **configuración de servicios de red** esenciales como DNS y DHCP en un entorno virtualizado funcional.

- La **automatización de tareas administrativas** mediante Ansible, lo que permitió una mayor eficiencia en la gestión de los sistemas.

- La **integración de un sistema de monitorización** basado en Grafana, Prometheus y Loki, que proporcionó métricas y logs centralizados para un control avanzado de los servicios.

A lo largo del proceso, se presentaron retos técnicos significativos, como problemas de espacio en disco, consumo de recursos en entornos virtualizados, e incompatibilidades entre herramientas, todos ellos resueltos mediante investigación autónoma y trabajo colaborativo.

Como conclusión, **M.A.S.T no solo cumple con los requisitos académicos del proyecto final de ciclo, sino que también representa una simulación realista de un entorno IT empresarial moderno**.

# Posibles ampliaciones y mejoras

En este apartado abordamos algunas ideas de ampliación y mejoras que podríamos implementar si decidiéramos seguir desarrollando este proyecto con vistas a convertirlo en una empresa de ciberseguridad. El objetivo sería diversificar nuestro catálogo de soluciones, mejorar la calidad técnica de nuestras herramientas y aumentar el nivel de personalización y control en nuestras implementaciones.

## Ampliaciones del catálogo

**USB auto inyectable (USB fingerprinting):**
Diseñar un dispositivo USB capaz de realizar fingerprinting del sistema donde se conecta, recolectando información básica del equipo automáticamente. Este tipo de herramienta es útil en auditorías de seguridad para obtener un reconocimiento inicial del entorno.

**Sistema de Detección de Intrusos (IDS):**
Desarrollar o integrar un IDS propio que analice el tráfico de red en tiempo real y alerte sobre posibles amenazas, comportamientos sospechosos o accesos no autorizados. Se podría integrar con los honeypots para mejorar la visibilidad del sistema.

**Implementación manual de honeypots personalizados:**
En lugar de depender de herramientas automatizadas como **T-Pot**, podríamos haber instalado manualmente distintos tipos de honeypots (SSH, HTTP, FTP, etc.), abriendo puertos y configurando servicios vulnerables en máquinas diseñadas desde cero. Esto nos daría un control más preciso sobre el comportamiento de los honeypots y nos permitiría adaptar la infraestructura a contextos específicos.

## Mejora de la infraestructura técnica

- **Personalización de logs y alertas:**
  Desarrollar un sistema de alertas personalizadas con integración en tiempo real (por ejemplo, mediante correo, Telegram o dashboards como Grafana).

- **Mejor gestión de los datos recolectados:**
  Utilizar bases de datos más eficientes y sistemas de análisis avanzado (como ELK Stack o Splunk) para procesar y visualizar los logs y eventos generados por los honeypots.

- **Implementación de alta disponibilidad:**
  Diseñar una arquitectura redundante para que, en caso de fallo de uno de los nodos, el sistema siga funcionando sin interrupciones.

### Otras líneas de trabajo

- **Integración con servicios de inteligencia de amenazas (Threat Intelligence):**
  Relacionar las IPs maliciosas detectadas con bases de datos públicas o privadas para enriquecer los datos recolectados.

- **Simulación de ataques reales controlados:**
  Utilizar herramientas como Metasploit o Cobalt Strike en entornos de laboratorio para evaluar el comportamiento de los honeypots y la efectividad de nuestras defensas.

# Agradecimientos

Una vez llegados a este punto, solo nos queda dar las gracias a todas las personas que, de una forma u otra, han brindado su apoyo, sus ánimos y su confianza para que este proyecto de final de curso se haya hecho realidad. Ha sido un camino largo, con esfuerzo, aprendizajes y muchos buenos momentos.

A continuación, cada integrante del equipo quiere expresar sus agradecimientos personales:

**Iván**→ Primero de todo, me gustaría dar las gracias a mis compañeros de equipo, por el esfuerzo, el compromiso y la paciencia que han demostrado durante todo el proyecto. Hemos pasado por algún que otro contratiempo, pero siempre supimos salir adelante como equipo, sin rendirnos.

También quiero agradecer profundamente a mi familia, que ha estado siempre pendiente de mí y del proyecto. En especial a mi padre, que se interesó desde el primer día por lo que hacíamos, preguntando y animándome a seguir. Gracias por acompañarme en este camino.

A mi novia, mil gracias por darme fuerzas en los momentos difíciles, por escucharme hablar horas y horas del proyecto sin quejarte, y por estar ahí siempre con palabras de ánimo cuando más las necesitaba.

A mis amigos, gracias por sacarme una sonrisa en los momentos de estrés, por hacerme desconectar cuando más lo necesitaba, y por ayudarme a ver las cosas con otra perspectiva.

Y por último, pero no menos importante, gracias al centro y a los profesores que nos han guiado durante todo el ciclo. Desde Juan Morote con sus míticas tablas de subnetting, hasta

Jordi Ferrero con su forma única de enseñarnos ciberseguridad de manera divertida y cercana. Gracias a todos por transmitirnos conocimiento, pasión y confianza.

Este proyecto me ha dejado un gran resultado, muchas risas, nuevos amigos, conocimientos valiosos y, sobre todo, una experiencia que recordaré siempre.

Mil gracias de verdad.

Mención honorífica: al señor de la cantina, simplemente gracias por crear esa maravilla llamada triángulos de chocolate.

**Rafa** → En primer lugar, quiero dar las gracias a mis compañeros Iván y Luis. Ha sido un placer trabajar con vosotros en este proyecto. Cada uno ha aportado su granito de arena y hemos sabido organizarnos incluso cuando las cosas se complicaban. Gracias por vuestro esfuerzo y por las risas que también han formado parte de todo esto.

También me gustaría agradecer a mi familia, que, aunque muchas veces no entendía exactamente qué estaba haciendo, me apoyaban igualmente porque confiaban en mí. Esa confianza ha sido fundamental para seguir adelante, incluso en los momentos más duros.

A mis amigos, que muchas veces se metían conmigo por ser de ciclo y no estar en bachillerato como ellos, y a los que yo a su vez también les devolvía las bromas porque ellos estaban todos los días estudiando. Esa complicidad y esas pequeñas peleas nos ayudaron a mantenernos unidos y a desconectar cuando más lo necesitábamos.

Quiero dar un agradecimiento muy especial a los profesores que nos han acompañado durante estos años:

- A **Caridad Castillo**, nuestra primera tutora al entrar al ciclo, que me ayudó en muchas ocasiones con sus tutorías y también con conversaciones cotidianas que siempre fueron de gran apoyo.

- A **Federico**, nuestro profesor de proyecto y miembro del equipo de orientación, quien me ayudó a aclarar mi ruta de estudios y me dio consejos muy valiosos sobre qué hacer.

- A **Víctor Cárceler**, con sus increíbles clases divertidas e interactivas, en las que aprendí mucho; además, a veces nos enseñaba ruso y otros temas que no formaban parte de la asignatura pero que resultaban realmente interesantes.

- A **José Camuñez**, nuestro tutor en el segundo año y profesor de sistemas operativos, recordado por sus míticas frases como “quiero soluciones, no problemas” o “recerca”, entre otras muchas.

- A **Jordi Ferrero**, quien, pese a sus difíciles exámenes, ha conseguido que me encante la ciberseguridad.

- A **Juan Morote**, el profesor de redes en nuestro primer año del ciclo y coordinador de Erasmus de nuestro centro, quien se esfuerza mucho porque aprendamos e hizo que me decantara más por ASIX para administrar redes.

- Y a todos los demás profesores que no he mencionado, pero a quienes también agradezco mucho por todo lo que nos han enseñado.

Este proyecto no solo ha sido un reto técnico, también ha sido una experiencia de crecimiento. Me voy con muchas cosas aprendidas, buenos recuerdos y la satisfacción de haberlo dado todo.

**Luis** →Quiero comenzar agradeciendo a mis compañeros de equipo, Iván y Rafa, por todo el trabajo, esfuerzo y dedicación que han puesto durante estos meses. Ha sido un camino largo, con momentos buenos y también momentos complicados, pero siempre supimos mantenernos unidos como equipo y salir adelante. Gracias por compartir esta experiencia conmigo.

También quiero agradecer a mi familia, en especial a mi madre, que siempre ha estado ahí preguntándome cómo iba todo, dándome ánimos para seguir adelante con el proyecto. Gracias por el apoyo constante, la paciencia y por confiar en mí.

A mis amigos y compañeros de clase por saber hacerme reír o sacarme sonrisas ya que soy alguien que le cuesta hablar con los demás pero gracias a ellos con sus bromas,chistes etc, he podido confiar más en mi mismo para comunicarme más con ellos.

Y, por supuesto, gracias a los profesores que me han acompañado a lo largo de este ciclo que me han hecho que me guste aún más la informática.

· Juan Morote

· Jordi Ferrero

· Jose Camuñez

· Victor Carceler

· Federico

Este proyecto ha sido un reto, pero también una gran oportunidad para aprender, superarme y compartir momentos inolvidables con todos mis compañeros. Gracias a todos por formar parte de este viaje.

# WEBGRAFIA:

https://www.youtube.com/watch?v=FtR9sFJlkSA&t=34s&pp=ygUXaG93IHRvIG1ha2UgYSBob25leSBwb3Q%3D

https://www.youtube.com/watch?v=SKhKNUo6rJU&t=191s&pp=ygUXaG93IHRvIG1ha2UgYSBob25leSBwb3Q%3D

https://www.youtube.com/watch?v=gI8LnMAhBv8&pp=ygUXaG93IHRvIG1ha2UgYSBob25leSBwb3Q%3D

https://es.wikipedia.org/wiki/Honeypot

https://www.incibe.es/empresas/blog/honeypot-una-trampa-para-los-ciberdelincuentes

https://grafana.com/

Manual para la creación del Honeypot

Video sobre la creación del Honeypot en Kali Linux

Viquipedia sobre las Inyecciones Creación de un HoneyPot Básico | Kali Linux |SQL

https://www.w3schools.com/w3css/default.asp

https://www.welivesecurity.com/la-es/2020/07/31/que-es-honeypot-como-implementarlo-nuestra-red/

https://www.youtube.com/watch?v=BDHKv2tRhFo

https://github.com/telekom-security/tpotce#running-in-a-vm

https://www.youtube.com/watch?v=uwnVK3DI8Sk

https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/10-beta/html/considerations_in_adopting_rhel_10/infrastructure-services?utm_source=chatgpt.com

https://www.redeszone.net/tutoriales/servidores/configurar-servidor-dns-bind-linux/?utm_source

https://kea.readthedocs.io/en/kea-2.2.0/arm/ddns.html?utm_source=chatgpt.com

https://verneacademy.com/blog/articulos-ciberseguridad/que-es-honeypot-y-para-que-sirve/?utm_source=chatgpt.com

https://www.startupdefense.io/es-us/blog/honeypots-una-guia-completa-sobre-senuelos-de-ciberseguridad?utm_source=chatgpt.com

https://cadenaser.com/nacional/2024/12/26/espana-nueva-capital-de-los-piratas-informaticos-israelies-cadena-ser/?utm_source=chatgpt.com

https://www.tarlogic.com/es/blog/empresas-de-ciberseguridad/?utm_source=chatgpt.com

https://es.wikipedia.org/wiki/Panda_Security?utm_source=chatgpt.com

https://www.paloaltonetworks.com/

https://www.fortinet.com/lat/resources/cyberglossary/types-of-cyber-attacks?utm_source=chatgpt.com

https://www.checkpoint.com/es/cyber-hub/cyber-security/what-is-cyber-attack/types-of-cyber-attacks/?

https://www.welivesecurity.com/es/

https://www.youtube.com/watch?v=rT_CjwKN380&t=22s → Cyberchef

https://www.youtube.com/watch?v=yw_KPBdAKHo → Spider foot

https://www.youtube.com/watch?v=n9mE5MXGkaA → Elasticsearch

https://www.spiderfoot.net

GitHub - SpiderFoot

https://www.elastic.co/elasticsearch

Documentación oficial

# ANEXO

## A.1. Listado de máquinas virtuales y direcciones IP

| MV | Servicio | Tarjeta | IP |
|---|---|---|---|
| 1 | DHCP & DNS | - enp0s3<br>- enp0s8 | - 192.168.12.218<br>- 192.168.18.1 |
| 2 | Pàgina web | - enp0s3 | - 192.168.12.250 |
| 3 | T-Pot | - enp0s3 | - 192.168.12.203 |
| 4 | Ansible | - enp0s3 | - 192.168.18.83 |
| 5 | Grafana | - enp0s3 | - 192.168.12.253:3000 |
| 6 | Prometheus | - enp0s3 | - 192.168.12.245: 9090 |
| 7 | Loki | - enp0s3 | - 192.168.12.247:3100 |

La máquina virtual de Prometheus se le tiene que agregar una ruta para poder llegar a la red 192.168.18.0:

sudo route add -net 192.168.18.0 netmask 255.255.255.0 gw 192.168.12.218

## A.2. Recursos técnicos utilizados

- **SO base:** Ubuntu Server 24.04
- **Entorno de virtualización:** VirtualBox
- **Plugins WordPress:** Elementor, Kubio, Spexo (versiones gratuitas)
- **Herramientas clave:** T-Pot, Grafana, Prometheus, Loki, Ansible, SpiderFoot, CyberChef, Bind9, KEA DHCP

## A.3. Problemas técnicos y soluciones

| Problema | Solución aplicada |
|---|---|
| Espacio insuficiente en T-Pot | Aumento de disco virtual y limpieza de logs residuales |
| Fallos de carga en Kibana | Revisión de RAM y swap, mejora del entorno gráfico del contenedor Docker |
| Incompatibilidades en WordPress | Establecer orden lógico entre plugins y evitar combinaciones inestables |
| Red no accesible desde Prometheus | Ruta estática añadida manualmente mediante `route add` |

## A.4. Descripciones

Ansible

Herramienta de automatización de código abierto que permite gestionar servidores, realizar configuraciones y desplegar aplicaciones a través de scripts en formato YAML llamados *playbooks*.

---

API (Application Programming Interface)

Interfaz de programación que permite la comunicación entre aplicaciones o servicios.

---

BDD (Base de Datos)

Sistema organizado para almacenar, gestionar y recuperar información digital. En este proyecto se usó MySQL para gestionar la base de datos del sitio web.

---

Bind9

Servicio de sistema de nombres de dominio (DNS) ampliamente utilizado en entornos Linux. Gestiona y resuelve nombres de dominio en redes privadas o públicas.

---

CLI (Command Line Interface)

Interfaz basada en texto que permite al usuario ejecutar comandos directamente mediante una terminal.

---

DHCP (Dynamic Host Configuration Protocol)

Protocolo de red que asigna automáticamente direcciones IP a dispositivos conectados. En este proyecto se utilizó KEA DHCP como servidor.

---

DNS (Domain Name System)

Sistema que traduce nombres de dominio legibles (como www.google.com) en direcciones IP. El servidor utilizado fue Bind9.

---

ELK Stack

Conjunto de herramientas compuesto por Elasticsearch, Logstash y Kibana, utilizado para la recolección, análisis y visualización de datos y logs.

---

GNU GPL (General Public License)

Licencia de software libre que garantiza el derecho a usar, modificar y distribuir software con libertad.

---

HTTP / HTTPS (HyperText Transfer Protocol / Secure)

Protocolos de transferencia de información en la web. HTTPS incluye una capa de cifrado mediante TLS/SSL.

---

IDS (Intrusion Detection System)

Sistema de detección de intrusos que monitorea tráfico y genera alertas ante comportamientos sospechosos.

---

IoT (Internet of Things)

Conjunto de dispositivos conectados a Internet que recopilan e intercambian datos, como sensores, cámaras, o dispositivos médicos.

---

IP (Internet Protocol)

Identificador único que cada dispositivo conectado a una red posee para su localización y comunicación.

---

KEA DHCP

Servidor DHCP moderno desarrollado por ISC, pensado para reemplazar al clásico *ISC DHCP*.

---

Kibana

Herramienta visual de la suite ELK que permite representar y explorar datos almacenados en Elasticsearch mediante paneles interactivos.

---

Loki

Sistema de agregación de logs desarrollado por los creadores de Grafana. Está optimizado para funcionar en conjunto con Prometheus y Grafana.

---

OSINT (Open Source Intelligence)

Información de fuentes públicas utilizada para investigaciones de seguridad informática. Herramientas como SpiderFoot automatizan este proceso.

---

Prometheus

Herramienta de recopilación de métricas orientada a series temporales. Recoge estadísticas de rendimiento de sistemas, servidores y servicios.

---

RAM (Random Access Memory)

Memoria de acceso aleatorio donde se cargan temporalmente datos y programas que el sistema está utilizando.

---

SCADA (Supervisory Control And Data Acquisition)

Sistemas de control industrial para monitorización y gestión remota de instalaciones físicas (como plantas eléctricas, fábricas o sistemas de agua).

---

SQL (Structured Query Language)

Lenguaje utilizado para interactuar con bases de datos relacionales, como MySQL, MariaDB o PostgreSQL.

---

SSH (Secure Shell)

Protocolo seguro para conectarse remotamente a otros equipos mediante una consola o terminal. Se utilizó para administrar el servidor de T-Pot.

---

SSL/TLS (Secure Socket Layer / Transport Layer Security)

Protocolos criptográficos que garantizan comunicaciones seguras en Internet, usados principalmente en HTTPS.

---

T-Pot

Plataforma de honeypots basada en contenedores Docker. Incluye múltiples servicios falsos para atraer y analizar ataques, como Cowrie, Dionaea, Galah, etc.

---

UDP / TCP

Protocolos de transporte utilizados para enviar datos entre dispositivos. TCP es orientado a conexión y confiable, mientras que UDP es más rápido pero menos seguro.

---

URL (Uniform Resource Locator)

Dirección web que especifica la localización de un recurso en Internet (por ejemplo, una página web).

---

VM (Virtual Machine)

Sistema operativo emulado que se ejecuta sobre otro sistema físico. Se usaron varias VMs para aislar los servicios.

---

VPN (Virtual Private Network)

Tecnología que permite crear una red privada segura sobre una red pública como Internet, cifrando los datos de transmisión.

---

WordPress

Sistema de gestión de contenido (CMS) utilizado para construir y administrar el sitio web del proyecto. Se complementó con plugins como Elementor y Kubio.

---

YAML (YAML Ain’t Markup Language)

Formato legible por humanos para definir configuraciones, utilizado principalmente en Ansible para definir tareas en *playbooks*.

---

Zeek

Plataforma para el análisis profundo de tráfico de red, utilizada en entornos de seguridad para detectar amenazas.

---

Elasticsearch

Es un motor de búsqueda y análisis distribuido basado en Lucene. Permite almacenar, buscar y analizar grandes volúmenes de datos en tiempo real. Es ampliamente utilizado en aplicaciones de log management, business analytics, monitoreo de seguridad y más.

---

Kibana

Es una herramienta de visualización que se usa en conjunto con Elasticsearch. Permite crear dashboards interactivos, visualizar datos en gráficos, mapas y tablas, y explorar logs de manera sencilla. Es parte del Elastic Stack (Elasticsearch, Logstash, Kibana).

---

WebUI

No es un producto específico, sino un término general que se refiere a una interfaz gráfica de usuario basada en la web. Puede ser cualquier aplicación o plataforma que proporcione una forma interactiva de acceder y gestionar servicios, como Kibana para Elasticsearch.

---

Beelzebub

Es un framework de honeypot de código abierto diseñado para detectar y analizar amenazas cibernéticas mediante simulaciones realistas con IA.

---

Galah

Es un honeypot web que usa modelos de lenguaje avanzados para generar respuestas dinámicas y engañar a los atacantes. Ambos forman parte de T-Pot 24.04.1, una plataforma que combina múltiples honeypots para mejorar la detección de amenazas.